ISSN 2288-4866 (Print)
ISSN 2288-4882 (Online)
http://www.jiisonline.org
J Intell Inform Syst 2019 December: 25(4): 105~122 http://dx.doi.org/10.13088/jiis.2019.25.4.105

문장 분류를 위한 정보 이득 및 유사도에 따른 단어 제거와 선택적 단어 임베딩 방안*

이민석
가톨릭대학교 경영학전공
(*ava.vi.riyenas@gmail.com*)

양석우
가톨릭대학교 심리학전공
(*josh.yang950@gmail.com*)

이홍주
가톨릭대학교 경영학전공
(*hongjoo@catholic.ac.kr*)

텍스트 데이터가 특정 범주에 속하는지 판별하는 문장 분류에서, 문장의 특징을 어떻게 표현하고 어떤 특징을 선택할 것인가는 분류기의 성능에 많은 영향을 미친다. 특징 선택의 목적은 차원을 축소하여도 데이터를 잘 설명할 수 있는 방안을 찾아내는 것이다. 다양한 방법이 제시되어 왔으며 Fisher Score나 정보 이득(Information Gain) 알고리즘 등을 통해 특징을 선택 하거나 문맥의 의미와 통사론적 정보를 가지는 Word2Vec 모델로 학습된 단어들을 벡터로 표현하여 차원을 축소하는 방안이 활발하게 연구되었다. 사전에 정의된 단어의 긍정 및 부정 점수에 따라 단어의 임베딩을 수정하는 방법 또한 시도하였다.

본 연구는 문장 분류 문제에 대해 선택적 단어 제거를 수행하고 임베딩을 적용하여 문장 분류 정확도를 향상시키는 방안을 제안한다. 텍스트 데이터에서 정보 이득 값이 낮은 단어들을 제거하고 단어 임베딩을 적용하는 방식과, 정보이득 값이 낮은 단어와 코사인 유사도가 높은 주변 단어를 추가로 선택하여 텍스트 데이터에서 제거하고 단어 임베딩을 재구성하는 방식이다.

본 연구에서 제안하는 방안을 수행함에 있어 데이터는 Amazon.com의 'Kindle' 제품에 대한 고객리뷰, IMDB의 영화리뷰, Yelp의 사용자 리뷰를 사용하였다. Amazon.com의 리뷰 데이터는 유용한 득표수가 5개 이상을 만족하고, 전체 득표 중 유용한 득표의 비율이 70% 이상인 리뷰에 대해 유용한 리뷰라고 판단하였다. Yelp의 경우는 유용한 득표수가 5개 이상인 리뷰 약 75만개 중 10만개를 무작위 추출하였다. 학습에 사용한 딥러닝 모델은 CNN, Attention-Based Bidirectional LSTM을 사용하였고, 단어 임베딩은 Word2Vec과 GloVe를 사용하였다. 단어 제거를 수행하지 않고 Word2Vec 및 GloVe 임베딩을 적용한 경우와 본 연구에서 제안하는 선택적으로 단어 제거를 수행하고 Word2Vec 임베딩을 적용한 경우를 비교하여 통계적 유의성을 검정하였다.

주제어 : 문장 분류, 특징 선택, 정보 이득, 단어 유사도, 단어 임베딩

논문접수일 : 2019년 6월 23일 논문수정일 : 2019년 9월 22일 게재확정일 : 2019년 11월 16일
원고유형 : 일반논문 교신저자 : 이홍주

1. 서론

텍스트 마이닝에서 특징을 나타내는 단어의 선택은 분류 모델의 성능에 상당한 영향력을 가진다(Lee and Lee, 2016; Lee and Lee, 2017). 단어가 증가할수록 처리해야할 계산의 양은 기하급수적으로 증가하지만 분류 정확도 증가는 한정적일 수 있다. 결과적으로 계산 비용의 증가와 모델의 과적합을 야기할 가능성이 높아진다(Li et al., 2016). 이를 개선하기 위해 전처리 과정에

* 이 논문은 2017년 대한민국 교육부와 한국연구재단의 지원을 받아 수행된 연구임 (NRF-2017S1A3A2066740).

서 데이터의 차원 축소가 많이 활용된다. 또한, 차원 축소를 위한 다양한 방법들이 제시되어 왔고, 단순히 데이터의 노이즈를 줄이는 것에서부터 통사론적 정보와 문맥을 함께 고려하면서 성능을 높이는 방안을 활용하였다.

문장 분류 문제는 텍스트의 내용을 분석하여 몇 가지의 범주에 할당하는 과정이며 다양한 자연어 처리 문제 중 하나이다. 기존의 문장 분류를 하는 과정에서 주성분 분석(Principle Component Analysis)(Jolliffe, 2002), 선형 판별 분석(Linear Discriminant Analysis)(Mika et al., 1999) 같은 특징 추출 알고리즘, Fisher Score(Duda et al., 2012), 정보 이득(Information Gain)(Lewis, 1992) 같은 특징 선택 알고리즘, 사전에 학습된 단어 임베딩을 통해 데이터를 저차원의 형태로 변환하는 임베딩 방식은 선행 연구에서 제시되었다. 예를 들어, 현재 자연어 처리 문제에서 활발하게 쓰이고 있는 단어 임베딩 방식은 Word2Vec (Mikolov et al., 2013), GloVe(Pennington et al., 2014), fastText(Bojanowski et al., 2016), ELMo (Peters et al., 2018) 등이 있다. 일반적으로 현존하는 단어 임베딩 방식은 의미 지향적이고, 라벨링(Labeling) 되어 있지 않은 데이터에서 문맥과 통사론적 정보를 넘어서는 충분한 정서적 정보는 파악하기 어렵다. 최근에는 성능 개선을 위해 단어 정제 모델을 통해 사전에 학습된 단어 임베딩을 수정하는 방법(Yu et al., 2017), 특징 선택 방식을 통해 선택된 단어의 유사 단어를 사용하여 기존의 단어 사전에 추가하는 방법 등이 시도되었다(Zhu et al., 2017).

본 연구는 문장 분류 문제에 대해 문장의 특징을 선택함에 있어 상대적으로 중요하지 않은 단어를 제거 후, 단어 임베딩을 생성하는 점에서 기존 연구와 차이가 있다. 상대적으로 덜 중요한 단어들을 분류하는 방안으로 Zhang and Tran (2011)이 제안한 정보 이득(Information Gain)을 활용하였으며 유사한 단어 선택에 코사인 유사도(Cosine Similarity)를 활용하였다. 본 연구는 정보 이득 값이 낮은 단어를 문장에서 제거하거나, 정보 이득 값이 낮은 단어와 코사인 유사도가 높은 단어를 추가로 제거하여 단어 임베딩을 구성하는 두 가지 방법론을 제안한다. 정보 이득 알고리즘을 통해 단어의 정보 이득 값을 구하고 정보 이득 값이 낮은 단어를 텍스트 데이터에서 제거하여 단어 임베딩을 구성하였다. 정보 이득 값이 낮은 단어와 코사인 유사도가 높은 단어를 추가로 선택하여 텍스트 데이터에서 제거하고 이를 활용하여 임베딩을 구성하는 방법도 함께 시도하였다. 본 연구에서 제안한 방법론을 평가하기 위해 단어 제거를 수행하지 않고 Word2Vec과 GloVe 임베딩을 적용한 경우와 선택적으로 단어 제거를 수행하고 Word2Vec 방식으로 단어 임베딩을 적용한 방식과 성능 비교를 하였다. 또한 실험 결과를 양측 검정으로 통계적 유의성을 확보하여 두 가지의 방법 중 적어도 한 가지의 방법에서 기존의 단어 임베딩 방식보다 본 연구에서 제안한 임베딩 방식이 통계적으로 유의함을 보였다. 본 연구를 진행함에 있어 데이터로 Amazon.com Kindle 카테고리 제품에 대한 리뷰, 영화 리뷰인 IMDB, 사용자 리뷰인 Yelp를 사용하였고, 모델은 Convolutional Neural Network (Kim, 2014)와 Attention-Based Bidirectional LSTM (P Zhou, 2016)을 사용하였다.

전반적인 논문의 내용은 다음과 같다. 2장에서 차원을 축소하기 위해 기존에 제안되었던 방안에 대해 기술하고, 3장에서 본 연구에서 제안하는 새로운 알고리즘에 대해 기술하였다. 4장에서 전반적인 실험 설계에 대하여 설명하고, 5장

에서 실험 결과를 제시하였다. 6장에서 내용을 종합한 결론을 제시하였으며 한계점에 대해 기술하였다.

2. 관련 연구

문장 분류의 목적은 사전에 정의된 범주에 문장을 분류하는 것이다. 문장 분류 문제에 감성 분석, 토픽 라벨링, 스팸 탐지 등이 포함된다. 문장 분류에서 성능을 높이기 위해 다양한 방법이 시도되어 왔으며, 기계 학습 방안들이 많이 적용되었다. Naïve Bayes(Sahami, 1996; Lewis, 1998)와 Support Vector Machine(Joachims, 1998) 방법들이, 차원 축소와 함께 적용되었을 때 더 높은 성과를 보였다(Mohan and Paramasivam, 2017; Kim, 2005). 차원 축소는 문장 분류 문제에서도 성능 향상과 모델의 과적합을 방지하기 위해 중요한 부분이며, 본 연구는 단어의 차원 축소에 초점을 맞추어 연구를 진행하였다. 지금까지 적용되어온 차원 축소의 방법은 크게 특징 추출, 특징 선택, 단어 임베딩으로 분류할 수 있다.

2.1 특징 추출 (Feature Extraction)

차원 축소에 대한 연구는 다양한 측면에서 진행되었고, 차원 축소 방법론 중 특징 추출 방식은 기존에 학습된 알고리즘에 적용 가능한 특징들이 없을 경우에 선호된다(J Li et al., 2016). 특징 추출은 고차원 데이터의 특징을 선형 혹은 비선형의 결합으로 보다 낮은 차원으로 투영시키는 방식이다.

선형 판별 분석(Mika et al., 1999)은 데이터를 한 특정 축에 투영(Projection)하고 그 데이터 내 군집을 잘 분류할 수 있는 직선을 찾는 것을 목적으로 한다. 선형 판별 분석의 핵심은 군집 간의 분산은 최대, 군집 내의 분산은 최소가 되도록 하는 벡터를 활용하여 데이터를 분류하는 직선을 구성하는 것이다. 군집의 정보를 보존하며 차원을 축소하는 선형 판별 분석과 달리 전체적인 분포를 고려하는 주성분 분석(Jolliffe, 2002)은 기존의 변수를 선형 결합하여 새 변수를 만들어내고, 데이터의 좌표 성분들 사이의 공분산 값을 원소로 취하는 공분산 행렬을 만들어 고유 값을 찾는다. 주성분 분석은 데이터 분포에서 특정 방향으로 분산이 큰 주성분을 찾아 데이터의 변화량이 가장 큰 축을 찾는다. 그 결과 고차원의 데이터를 저차원 공간으로 변환시켜 분석을 용이하게 한다.

Locally Linear Embedding(Roweis and Saul, 2000)은 특정 축에 데이터를 투영하는 주성분 분석과 달리 매니폴드(Manifold) 학습방식을 취하는 비선형 차원 축소 기법이다. LLE 알고리즘은 서로 인접한 데이터를 보존하고 입력 데이터를 저차원의 단일 글로벌 좌표계(Single Global System)에 매핑한다. 그 과정은 각 데이터에서 이웃을 구하고 가중치 행렬을 구성하여 이 가중치를 유지하며 저차원의 형태로 변환하는 형식이다. LLE의 장점은 기본적으로 고차원의 데이터를 저차원으로 매핑이 가능하며 다루기 쉽고 비선형 임베딩이 가능하다는 점이다.

2.2 특징 선택 (Feature Selection)

특징 선택 알고리즘은 자연어 처리 문제에서 어떠한 특징이 문제 해결에 있어 중요한 것인지 판별하는데 활용된다. Fisher Score(Duda et al., 2012)는 유사성을 기반으로 다른 군집 간의 분산

이 크고, 군집 내의 개별 데이터들 간의 분산은 작은 특징을 찾는다. Fisher Score가 높다면 한 특징으로 분류한 군집 간의 분산이 크다는 것을 의미하고 해당 특징은 분류를 하는데 있어 유용한 기준이 될 수 있다.

희소한 정보는 그렇지 않은 정보에 비해 더 유익한 정보라는 정보 이론(Information Theory)은 현존하는 많은 특징 선택 알고리즘의 근간이 된다(J Li et al., 2016). 정보 이득(Lewis, 1992)은 데이터의 혼잡도에 대한 엔트로피 개념을 통해 계산할 수 있다. 정보 이득 값은 상위 노드와 하위 노드의 엔트로피 차이로 계산되며 그 값이 높을수록 정보량이 많아 데이터를 잘 구분함을 의미한다.

Minimum-redundancy-maximum-relevance(mRMR)(Peng et al., 2005) 또한 정보 이론을 기초로 하며 데이터의 불필요한 중복을 줄이고, 데이터 간의 상관성을 높이는 것을 목적으로 한다. 데이터의 중복성과 상관성은 피어슨 상관계수와 정보 이득을 통해 정의되고, 탐욕 알고리즘(Greedy Algorithm)을 통해 특징을 선택한다. mRMR은 특징 선택을 함으로써 데이터 중복성을 감소시키고 독립변수 간의 관계를 더욱 강하게 하여 데이터의 해석력을 증대한다.

2.3 단어 임베딩 (Word Embedding)

단어 임베딩은 분배적 가설(Distributional Hypothesis)(Sahlgen, 2008)을 기반으로 단어를 벡터로 변경하는 방안이다. Word2Vec은 학습 방안에 따라 CBOW와 Skip-gram 모델이 있는데 Skip-gram이 CBOW에 비해 전체적으로 좋은 결과를 보이고 있다(Mikolov et al., 2013). Skip-gram은 중심 단어에 대해 주변 단어가 나타날 확률을 최대화 하는 방향으로 학습을 진행한다. 또한 중심 단어와 주변 단어 벡터의 내적이 코사인 유사도가 되도록 단어 벡터를 임베딩한다. 현재 Word2Vec에서 제안된 Skip-gram, negative sampling 방식은 자연어 처리 영역을 넘어 컴퓨터 비전(Frome, Corrado, and Shlens, 2013), 협업 필터링(Barkan and Koenigstein, 2016)과 같은 다양한 분야에 적용되고 있다(Barkan, 2017).

GloVe(Global Vectors for Word Representation)는 잠재의미분석(LSA)과 Word2Vec의 장점만을 반영하여 기존의 방법의 단점을 보완하였다. LSA는 선행 연구에서 밝힌 것과 같이 말뭉치 전체의 통계적 정보를 활용하고(Landauer and Dumais, 1997), 단어와 문맥 간의 내재적인 의미를 보존하여 모델 성능에 도움을 주며(Deerwester et al, 1990; Landauer and Dumais, 1997), 입력 데이터의 노이즈 제거(Rapp, 2003) 등에 효과가 있지만 단어 혹은 문서 간의 유사도 측정에는 어려움이 있다. 반면 Word2Vec은 단어 간의 유사도 측정이 가능하지만 연구자가 지정한 윈도우 규모 내에서 학습을 하기 때문에 말뭉치 전체의 통계적 정보를 반영하기 어렵다. 결과적으로 GloVe는 두 단어의 벡터 내적이 유사도가 아니라 동시 출현(co-occurrence) 확률의 로그 값이 되도록 목적함수를 갖는다.

fastText(Bojanowski et al., 2016)는 Skip-gram 방식을 사용하며 전반적으로 Word2Vec과 유사하지만 단어를 임베딩 하는 방식에서 차이가 있다. 단어를 개별 단어가 아닌 n-gram의 부분 단어로 임베딩 하여 희소 단어, 오타 등 비정형 데이터와 같은 노이즈가 많은 말뭉치 처리에 강점을 갖는다.

마지막으로 ELMo(Embedding from Language Models)는 기존에 제안된 임베딩 방식인 Word2Vec, GloVe, fastText와 달리 사전에 학습된 Bidirectional

LSTM을 사용하였다는 점에서 차이가 있다 (Peters et al., 2018). 앞서 제안된 방법들은 단어의 동시 출현 정보를 활용하기 때문에 다른 단어임에도 불구하고 비슷한 벡터 값을 가질 경우 유사한 의미를 지니는 것으로 임베딩 되었다. 하지만 ELMo는 Bidirectional LSTM 신경망을 통해 각 단계의 은닉층들을 조합하여 문맥을 고려하며 단어를 임베딩 하였으며, 같은 단어라 할지라도 문맥에 따라 다른 벡터 값을 가지게 임베딩 된다(Peters et al., 2018).

3. 제안 알고리즘

본 연구는 문장의 특징을 선택하고 단어 임베딩하는 과정에서 분류 성과 증대를 위해 두 가지 방안을 제안한다. 첫 번째 방안은 정보 이득 알고리즘을 사용하여 정보 이득 값이 낮은 단어를 파악한 뒤, 정보 이득 값이 상대적으로 낮은 단어들을 텍스트 데이터에서 제거하고 단어 임베딩을 구성하는 것이다. 두 번째 방안은 첫 번째 방안에서 선택된 정보 이득 값이 낮은 단어와 그 주변에 있는 유사 단어들을 텍스트 데이터에서 함께 제거하여 단어 임베딩을 구성하는 것이다.

3.1 정보 이득

엔트로피는 주어진 데이터 집합의 혼잡도를 말하며 서로 다른 종류의 데이터가 섞여 있으면 혼잡도가 높은 것으로 간주한다. 정보 이득은 상위 노드와 하위 노드 간 엔트로피의 차이를 구하고 이를 해당 속성의 변별력으로 간주한다. 본 연구에서는 텍스트 데이터 내 단어들의 엔트로피를 통해 정보 이득 값을 구한다. 단어의 정보 이득 값이 낮을수록 해당 단어가 문장 분류에 있어 변별력이 낮다는 것을 의미하므로, 정보 이득 값이 낮은 단어를 제거하였다. 전체 문서 중 특정 문서가 클래스 s_i에 속할 확률을 $P(s_i)$라고 하면, 전체 문서의 엔트로피는 $H(s)$로 정의되고 단어 w가 존재하는 경우에 대한 문서 엔트로피 $H(s \mid w)$는 다음과 같이 계산된다.

$$H(s) = -\sum_{i=1}^{q} \mathrm{P}(s_i) \log_2 \mathrm{P}(s_i)$$

$$H(s \mid w) = -\sum_{i=1}^{q} \mathrm{P}(s_i \mid w) \log_2 \mathrm{P}(s_i \mid w)$$

$\mathrm{P}(w)$는 전체 문서 중 단어 w를 포함한 문서의 확률이고, $P(s_i \mid w)$는 단어를 포함한 문서가 s_i 카테고리에 속할 확률이다. 반대로 $P(\widehat{w})$은 전체 문서 중 단어 w를 포함하지 않은 문서의 확률이며, $P(s_i \mid \widehat{w})$은 단어 w를 포함하고 있지 않은 리뷰가 s_i카테고리에 속할 확률이다. 따라서 단어 w가 존재하는 경우와 그렇지 않은 경우에 대해 단어 w의 정보 이득은 다음과 같이 계산된다 (Zhang and Tran, 2011).

$$Gain(w) = -\sum_{i=1}^{q} \mathrm{P}(s_i) \log_2 \mathrm{P}(s_i)$$

$$+ \mathrm{P}(w) \sum_{i=1}^{q} \mathrm{P}(s_i \mid w) \log_2 \mathrm{P}(s_i \mid w)$$

$$+ \mathrm{P}(\widehat{w}) \sum_{i=1}^{q} \mathrm{P}(s_i \mid \widehat{w}) \log_2 \mathrm{P}(s_i \mid \widehat{w})$$

본 연구에서는 리뷰의 평점에 따라 긍정/부정 리뷰로 분류하거나 리뷰의 유용 투표수에 따라

긍정(s_1)과 부정(s_2)으로 분류하였고 최종 정보 이득 값은 다음과 같다.

$$Gain(w) = \begin{cases} Gain(w) & if\ \mathrm{P}(s_1|\,w) < \mathrm{P}(s_2|\,w) \\ -Gain(w) & otherwise \end{cases}$$

정보 이득 알고리즘을 통해 텍스트 데이터에서 제거할 단어의 개수를 n이라 할 때, 제거 대상이 되는 단어의 집합은 $Drop\,Word_{(n)}$으로 정의할 수 있다. 제거 단어의 수 n은 정보 이득 값이 낮은 하위 30, 50, 100, 150개로 설정하였다.

$$DW_{(n)} = \{\ I_1,\ I_2,\ \cdots\ I_{n-1},\ I_n\ \}\quad \mathrm{n} = (30, 50, 100, 150)$$

3.2 단어 유사도

정보 이득 값이 낮은 단어 주변에 있는 단어를 찾기 위해 코사인 유사도를 사용하였다. 코사인 유사도는 내적 공간의 두 벡터 간 각도의 코사인 값을 이용하여 측정된 벡터 간의 유사도로 정의되며 단어 A와 B의 유사도는 다음과 같이 측정된다.

$$Similarity(A,B) = \cos\theta = \frac{\mathrm{A}\cdot\mathrm{B}}{\|\mathrm{A}\|\cdot\|\mathrm{B}\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n}(A_i)^2} \times \sqrt{\sum_{i=1}^{n}(B_i)^2}}$$

본 연구는 3.1절의 정보 이득 알고리즘을 통해 정보 이득 값이 낮은 n 개의 단어를 선택하고, n 개의 단어에 대해 코사인 유사도가 높은 t개의 단어를 추가로 선택하여 총 n + (n*t)개의 단어를 추출하였다. 두 단어(A, B)의 유사도 측정을 위해 실험에 사용한 텍스트 데이터를 Word2Vec 방식으로 만든 100차원 단어 임베딩에서 각 단어에 대해 100차원의 벡터를 사용하여 두 단어 벡터의 내각을 구함으로써 유사도를 도출하였다. 3.1절에서 정보 이득 값이 낮은 n개의 제거 대상이 되는 단어의 집합을 $Drop\,Word_{(n)}$으로 정의 했으므로 주변 t개의 단어를 포함한 결과는 다음과 같은 행렬 $Drop\,Word_{(n,t)}$로 나타낼 수 있다. n의 값은 동일하고 주변 단어의 개수는 5, 10, 15로 설정하였다. <Figure 1>은 정보 이득 값이 낮은 단어 I_n에 대해 주변의 t개의 단어를 추출한 예시이다.

$$\mathrm{n} = (30, 50, 100, 150),\qquad \mathrm{t} = (5, 10, 15)$$

$$DW_{(n,t)} = \begin{Bmatrix} I_1 & I_2 & \cdots & I_{n-1} & I_n \\ I_1S_1 & I_2S_1 & \cdots & I_{n-1}S_1 & I_nS_1 \\ I_1S_2 & I_2S_2 & \cdots & I_{n-1}S_2 & I_nS_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ I_1S_t & I_2S_t & \cdots & I_{n-1}S_t & I_nS_t \end{Bmatrix}$$

$I_n = author$

	word	info_gain
	even	−1.028067
	they	−1.029018
	into	−1.029129
	me	−1.029166
	up	−1.029313
	what	−1.030473
	about	−1.030750
	just	−1.030770
	author	−1.031285
	it	−1.031738
	t	−1.031876
	so	−1.032636

Extract Similar Words from author (t = 10)

0	author	1.000000
1	ridener	0.591651
2	tijan	0.590288
3	authors	0.588544
4	backlist	0.580712
5	bolton	0.578490
6	preface	0.575716
7	kassanna	0.572547
8	jani	0.569824
9	jasinda	0.569023
10	osborn	0.564487

〈Figure 1〉 Example of Extracting Similar Words from I_n

3.3 단어 제거 및 임베딩 생성

3.1절과 3.2절을 통해 제거 대상이 되는 단어들을 $DW_{(n)}$과 $DW_{(n,t)}$라 할 때, 텍스트 데이터에서 해당 단어들을 제거하는 알고리즘은 <Figure 2>와 같다. 실험에 사용할 텍스트 데이터를 토큰으로 바꿔주는 전처리를 수행한 뒤 문장 내에 존

Algorithm 1. Token Elimination

Token = T_n

Input : Tokenized Sentences = [$T_1, T_2, T_3, \dots, T_n$]

Case 1.
Output : Dropped Sentences($DS_{(n)}$) = [ ]

for Token in Tokenized Sentences
  if Token not in $DW_{(n)}$
    Add Token to $DS_{(n)}$

Case 2.
Output : Dropped Sentences($DS_{(n,t)}$) = [ ]

for Token in Tokenized Sentences
  if Token not in $DW_{(n,t)}$
    Add Token to $DS_{(n,t)}$

〈Figure 2〉 Token Elimination Algorithm

재하는 임의의 토큰 T_n에 대해 T_n이 제거 대상이 되는 $DW_{(n)}$ 또는 $DW_{(n,t)}$에 존재하지 않는다면 $Drop\ Sentence_{(n)}$과 $Drop\ Sentence_{(n,t)}$에 각각 추가한다. 이 작업을 반복하여 최종적으로 제거 대상이 되는 단어 $DW_{(n)}$과 $DW_{(n,t)}$를 제거한 결과인 $DS_{(n)}$과 $DS_{(n,t)}$를 얻을 수 있다.

단어 제거 알고리즘을 통해 얻은 결과인 $DS_{(n)}$과 $DS_{(n,t)}$를 통해 단어 임베딩을 생성하면 그 결과는 $Drop\ Embedding_{(n)}$과 $Drop\ Embedding_{(n,t)}$로 정의할 수 있다.

4. 실험 및 결과

4.1 데이터 선정 및 분류

본 연구는 Amazon.com ‘Kindle’ 제품에 대한 사용자 리뷰, IMDB의 영화 리뷰, Yelp의 사용자 리뷰 데이터를 사용하였고 각 데이터의 크기와 분류 클래스는 <Table 1>과 같다. Kindle의 유용한 리뷰 선택은 전체 데이터에서 유용한 표(Helpful Vote)를 5개 이상 받고, 전체 득표(Total vote)에서 70% 이상 유용한 표를 받은 리뷰를 유용한 리뷰로 분류하였다. Yelp의 경우 유용한 리뷰 선택에 있어 전체 득표 항목을 알 수 없기 때문에 유용한 표를 5개 이상 받은 리뷰 약 750,000개 중 100,000개를 무작위 추출하여 유용한 리뷰로 분류하였다. Kindle과 Yelp의 유용하지 않은 리뷰는 유용한 표가 0인 리뷰들 중에서 유용한 리뷰의 개수만큼 무작위 추출하였다. 각 리뷰 데이터에서 학습 집합과 테스트 집합을 80:20으로 나누고 다시 학습 집합에서 80%를 학습에, 나머지 20%를 학습한 것을 검증하는데 사용하였다.

〈Table 1〉 Information about Review Data

Datasets	Description	Size	True Size	False Size	Number of Class
Kindle	Customer Review	63,532	31,766	31,766	2
IMDB	Movie Review	50,000	25,000	25,000	2
Yelp	User Review	200,000	100,000	100,000	2

4.2 단어 임베딩

단어 임베딩은 Python의 Word2Vec 라이브러리를 활용하여 생성하였고, 5회 이상 등장한 단어들을 가지고 100차원의 Skip Gram 방식을 사용하였다. 나머지 파라미터는 기본 설정 값을 사용하였다. $DS_{(n)}$과 $DS_{(n,t)}$를 통해 만든 단어 임베딩 $DE_{(n)}$과 $DE_{(n,t)}$는 <Table 2>와 같이 나타낼 수 있다. 이를 실험에 사용한 데이터에 적용하면 그 결과는 <Table 3>, <Table 4>와 같다. <Table 3>은 정보 이득 기반 단어 제거 방법을 적용한 결과이고, <Table 4>는 정보 이득 및 유사도 기반 단어 제거 방법을 적용한 결과이다. 제거 비율은 방법에 따라 하나의 리뷰 당 평균적으로 제거된 단어의 비율을 나타낸다. <Table 3>의 IMDB 데이터의 경우 제거 정도가 미미하여 괄

〈Table 2〉 Information about Drop Embeddings

$DS_{(n)}$ & $DE_{(n)}$		$DS_{(n,t)}$ & $DE_{(n,t)}$	
$DS_{(30)}$	$DE_{(30)}$	$DS_{(30,5)}$	$DE_{(30,5)}$
$DS_{(50)}$	$DE_{(50)}$	$DS_{(30,10)}$	$DE_{(30,10)}$
⋮	⋮	⋮	⋮
$DS_{(1500)}$	$DE_{(1500)}$	$DS_{(150,15)}$	$DE_{(150,15)}$

〈Table 3〉 Mean Dropped Ratio per Sentences in $DE_{(n)}$

	Embedding	Dropped Ratio(%)		Embedding	Dropped Ratio(%)		Embedding	Dropped Ratio(%)
Kindle	$DE_{(30)}$	29.05	IMDB	$DE_{(30)}$	0 (5)	Yelp	$DE_{(30)}$	24.11
	$DE_{(50)}$	38.05		$DE_{(50)}$	0 (5)		$DE_{(50)}$	30.71
	$DE_{(100)}$	50.13		$DE_{(100)}$	0 (5)		$DE_{(100)}$	47.46
	$DE_{(150)}$	56.69		$DE_{(150)}$	0 (5.11)		$DE_{(150)}$	52.5
	$DE_{(300)}$	65.83		$DE_{(300)}$	0 (7.77)		$DE_{(300)}$	60.43
	$DE_{(500)}$	71.23		$DE_{(500)}$	0 (7.77)		$DE_{(500)}$	66.39
	$DE_{(1000)}$	78.1		$DE_{(1000)}$	0.01 (8.02)		$DE_{(1000)}$	73.79
	$DE_{(1500)}$	81.57		$DE_{(1500)}$	0.01 (9.35)		$DE_{(1500)}$	77.86

〈Table 4〉 Mean Dropped Ratio per Sentences in $DE_{(n,t)}$

	Embedding	Dropped Ratio(%)		Embedding	Dropped Ratio(%)		Embedding	Dropped Ratio(%)
Kindle	$DE_{(30,5)}$	32.33	IMDB	$DE_{(30,5)}$	5.93	Yelp	$DE_{(30,5)}$	28.02
	$DE_{(30,10)}$	41.59		$DE_{(30,10)}$	7.58		$DE_{(30,10)}$	29.94
	$DE_{(30,15)}$	42.42		$DE_{(30,15)}$	7.64		$DE_{(30,15)}$	30.96
	$DE_{(50,5)}$	42.61		$DE_{(50,5)}$	6.66		$DE_{(50,5)}$	35.9
	$DE_{(50,10)}$	47.52		$DE_{(50,10)}$	7.44		$DE_{(50,10)}$	37.76
	$DE_{(50,15)}$	48.75		$DE_{(50,15)}$	11.51		$DE_{(50,15)}$	38.49
	$DE_{(100,5)}$	56.06		$DE_{(100,5)}$	12.35		$DE_{(100,5)}$	50.02
	$DE_{(100,10)}$	59.18		$DE_{(100,10)}$	12.19		$DE_{(100,10)}$	52.24
	$DE_{(100,15)}$	60.82		$DE_{(100,15)}$	13.79		$DE_{(100,15)}$	53.41
	$DE_{(150,5)}$	63.06		$DE_{(150,5)}$	11.66		$DE_{(150,5)}$	54.77
	$DE_{(150,10)}$	66.13		$DE_{(150,10)}$	12.97		$DE_{(150,10)}$	57.28
	$DE_{(150,15)}$	67.72		$DE_{(150,15)}$	19.58		$DE_{(150,15)}$	58

호 안에 리뷰 중 가장 많이 제거된 비율을 추가하였다.

4.3 학습 모델 및 평가 방식

데이터를 학습하는데 사용한 모델은 CNN과 Attention-Based Bidirectional LSTM 모델을 사용하였다. 4-1, 4-2절의 데이터에 위의 두 모델을 사용하여 <Table 5>에 나타난 네 가지 단어 임베딩을 적용하여 실험을 수행하였다. 전체 데이터셋을 학습 집합과 테스트 집합으로 무작위 층화 추출하여 나누었으며, 분류 성과는 정확도(accuracy)와 f1 스코어를 활용하여 측정하였다. 총 30회 실험을 반복하였으며, 방안 간의 성과 차이를 통계적으로 검증하기 위하여 반복수행된 성과를 가지고 독립표본 t 테스트를 수행하였다.

〈Table 5〉 Comparison with Embeddings

Data	Word Embedding
Raw Text Data	Word2Vec(100D)
Raw Text Data	GloVe(100D)
$DS_{(n)}$	$DE_{(n)}$
$DS_{(n,t)}$	$DE_{(n,t)}$

4.4 선택적 단어 임베딩 적용 결과

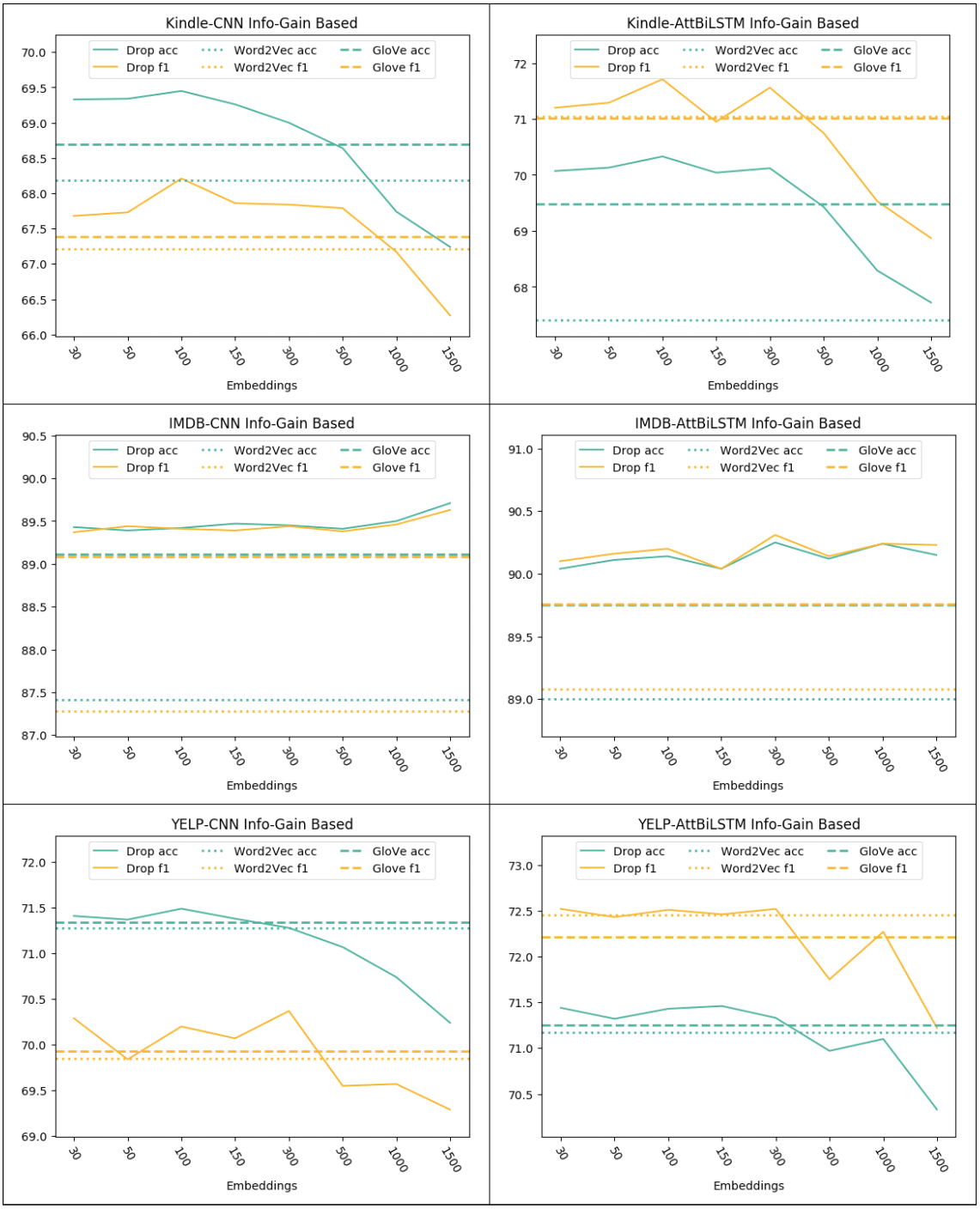
<Table 6>는 리뷰 데이터와 학습 모델을 활용하여 30회 반복 실험을 수행한 후의 성과의 평균값이다. 제안한 두 가지 방법에서 가장 좋은 성능을 보인 경우와 Word2Vec 및 GloVe를 활용한 경우와 t 테스트를 수행하여 통계적 유의성을 검정하였다. 실험 결과 모든 리뷰 데이터에 대해 본 연구에서 제안하는 방법론 중 적어도 한 가지 이상이 정확도와 f1 스코어에 대해 성과가 높았으며 통계적으로 유의한 차이를 나타냈다.

4.5 단어 제거 수에 따른 성능 변화

<Figure 3>는 정보 이득 기반 단어 제거 수에 따른 임베딩별 성능을 비교한 그래프이고, <Figure 4>는 정보 이득 및 코사인 유사도 기반 단어 제거 방법에 따른 임베딩별 성능을 비교한 그래프이다. 공통적으로 단어 제거 수가 증가함에 따라 성능이 좋아지다가 최적 제거 지점을 지난 이후 점점 떨어지는 추세를 보인다. 예외적으로 IMDB 리뷰에 CNN모델을 사용한 경우 정보 이득 기반 방법론을 적용하여 단어를 가장 많이 제거했을 때 성능이 가장 좋았고, 정보 이득 값이 낮은 단어뿐만 아니라 유사한 단어까지 함께 제거했을 때 오히려 결과가 떨어졌다. 성능이 떨어진 것은 Word2Vec이 단어의 동시 출현 정보를 보존하기 때문에 실제로 크게 관련이 없는 단어들이 가깝게 임베딩 되어 유사한 단어로 선택되고, 지워졌기 때문에 성능이 나빠진 것으로 보인다. 이를 종합하면 리뷰 데이터와 사용하는 모델, 그리고 본 논문에서 제안하는 방법론에 따라 단어를 제거하는 최적 값이 달라지므로 연구자가 사용하는 데이터 및 모델에 따라 제거하는 단어 수의 튜닝이 필요하다.

〈Table 6〉 Comparison with Word2Vec, GloVe, $DE_{(n)}$, $DE_{(n,t)}$ results and t-test.

Dataset	Model	Embedding	Acc	f1	t-test with Base		t-test with GloVe	
					Acc	f1	Acc	f1
Kindle	CNN	Word2Vec	68.18 (0.2)	67.21 (1.2)				
		GloVe	68.7 (0.5)	67.39 (1.5)				
		$DE_{(100)}$	***69.45 (0.3)***	68.21 (1.3)	significant (0)	significant (0)	significant (0)	significant (0.031)
		$DE_{(50,10)}$	69.31 (0.4)	***68.69 (1)***	significant (0)	significant (0.002)	significant (0)	significant (0)
	AttBiLSTM	Word2Vec	67.41 (0.2)	71.04 (0.5)				
		GloVe	69.48 (0.5)	71.01 (0.9)				
		$DE_{(100)}$	70.33 (0.4)	71.71 (0.5)	significant (0)	significant (0)	significant (0)	significant (0)
		$DE_{(100,5)}$	***70.8 (0.4)***	***72.4 (0.4)***	significant (0)	significant (0)	significant (0)	significant (0)
IMDB	CNN	Word2Vec	87.41 (0.5)	87.28 (0.7)				
		GloVe	89.11 (0.4)	89.08 (0.5)				
		$DE_{(1500)}$	***89.71 (0.4)***	***89.63 (0.5)***	significant (0)	significant (0)	significant (0)	significant (0)
		$DE_{(30,5)}$	88.82 (0.3)	88.79 (0.3)	significant (0)	significant (0)	significant (0)	significant (0)
	AttBiLSTM	Word2Vec	89 (0.5)	89.08 (0.3)				
		GloVe	89.75 (0.5)	89.76 (0.5)				
		$DE_{(300)}$	***90.25 (0.1)***	***90.31 (0.1)***	significant (0)	significant (0)	significant (0)	significant (0)
		$DE_{(30,5)}$	89.37 (0.3)	89.4 (0.3)	significant (0)	significant (0)	significant (0)	significant (0)
Yelp	CNN	Word2Vec	71.28 (0.4)	69.85 (1)				
		GloVe	71.34 (0.3)	69.93 (1.1)				
		$DE_{(100)}$	***71.49 (0.2)***	70.2 (0.8)	significant (0)	insignificant (0.117)	significant (0.035)	insignificant (0.277)
		$DE_{(50,10)}$	71.12 (0.2)	***70.4 (0.7)***	insignificant (0.069)	significant (0.014)	significant (0)	significant (0.038)
	AttBiLSTM	Word2Vec	71.17 (0.7)	72.45 (0.6)				
		GloVe	71.25 (0.2)	72.21 (0.6)				
		$DE_{(150)}$	***71.46 (0.1)***	72.46 (0.6)	significant (0)	insignificant (0.95)	significant (0)	insignificant (0.13)
		$DE_{(50,10)}$	71.3 (0.7)	***72.75 (0.5)***	insignificant (0.104)	significant (0.035)	insignificant (0.222)	significant (0)



〈Figure 3〉 Comparison with Information Gain base Dropped Embeddings

〈Figure 4〉 Comparison with Information Gain & Cosine Similarity base Dropped Embeddings

5. 결론

본 연구는 문장 분류에서 문장의 특징을 어떻게 선택할 것인 지가 분류 모형의 성능에 많은 영향을 미치기에, 선택적으로 단어 제거를 수행하고 임베딩을 적용하여 문장 분류 정확도를 향상시키는 두 가지 방안을 제안하였다. 텍스트 데이터에서 정보 이득 값이 낮은 단어들을 선택하여 문장에서 제거하거나, 해당 단어와 코사인 유사도가 높은 단어들을 함께 제거하여 문장의 특징을 선택하고 단어 임베딩을 생성한 점에서 기존 연구와 차별점을 갖는다. 제거한 텍스트 데이터를 바탕으로 Word2Vec을 통해 단어 임베딩을 만들고 이를 딥러닝 모델에 적용하여 선택적으로 단어 제거를 수행하지 않은 경우와 성능을 비교하였다.

제안한 방법론을 평가하기 위한 데이터로 사용자 리뷰와 영화 리뷰 데이터를 사용하였고, 딥러닝 모델로 CNN과 Attention-Based Bidirectional LSTM을 사용하였다. 실험 결과 제안한 방법론을 통해 선택적으로 단어 제거를 수행하고 임베딩을 적용한 경우가 그렇지 않은 경우에 비해 모든 결과에서 적어도 하나 이상이 통계적으로 유의한 차이가 있음을 보였다. 제안한 방법론을 적용함에 있어 연구자의 연구 목적과 사용하는 데이터 및 모델에 따라 어떤 방법론을 사용할지, 단어를 얼마나 제거할지가 달라짐을 확인할 수 있었다.

실무적 관점에서 제안한 방법론은 다음과 같이 적용할 수 있다. 본 연구는 사용자들이 작성한 리뷰 데이터를 활용하여 리뷰의 유용여부나, 긍정/부정 같은 감성 분류를 판별함에 있어 분류 정확도를 높이는 방법론을 제안하였다. 이는 서비스 기획 및 마케팅 관련 종사자로 하여금 자사가 제공하는 재화나 서비스에 대해 소비자들이 어떻게 인지하고 있는지 파악할 수 있는 자료로 활용 가능하다. 뿐만 아니라 유용하지 않은 리뷰를 판별하여 해당 리뷰를 덜 노출시켜 소비자로 하여금 바람직한 구매의사결정을 내리는데 기여할 수 있다. 문장 분류 엔진을 설계하는 엔지니어들은 본 연구 결과를 활용하여 기존 엔진의 성과를 높이는 방안을 시도해 볼 수 있다.

본 연구에서 제안하는 방법론의 성능을 평가하기 위해 텍스트 데이터에서 숫자 및 특수문자를 제거하는 등 최소한의 전처리를 수행하였으나 보다 심층적인 전처리를 수행한 뒤 제안하는 방법론을 적용하면 지금보다 성능 개선의 여지가 있을 것이라 생각한다. 또한, 유사한 단어를 찾기 위해 사용한 Word2Vec은 단어의 동시 출현 정보를 저장하여 단어 임베딩을 만들기 때문에 코사인 유사도로 측정한 유사 단어가 실제로는 유사한 단어가 아닌 경우가 존재한다. 따라서 보다 정밀하게 유사한 단어를 측정하는 방안의 적용이나 연구가 필요하다. Word2Vec이 아닌 다른 단어 임베딩에 대해 본 연구에서 제안하는 방법론을 적용했을 때 성능이 개선되는지는 확인하지 못했다. 그렇기에 추후 다른 단어 임베딩 방안에 대해 선택적으로 단어 제거를 수행하고 임베딩을 생성하는 방법을 적용하는 연구를 수행할 필요가 있다.

참고문헌(References)

Azhagusundari, B. and A.S. Thanamani, “Feature Selection based on Information Gain,” *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, Vol.2,

No.2(2013), 18- 21.

Barkan, O., "Bayesian Neural Word Embedding," *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)*, (2017)

Barkan, O. and N. Koenigstein.“Item2Vec: Neural Item Embedding for Collaborative Filtering,” arXiv Preprint arXiv:1603.04259 (2016).

Bojanowski, P., E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *CoRR* abs/1607.04606, (2016)

Deerwester, S., S.T. Dumais, T.K. Landauer, G.W. Furnas, and R. Harshman. "Indexing by latent semantic analysis," *Journal of the American Society of Information Science*, Vol.41, No.6(1990), 391~407.

Duda, R.O., P.E. Hart, and D.G. Stork. *Pattern classification*, Wiley, 2000.

Frome, A., G. Corrado, and J. Shlens, “Devise: A Deep Visual-Semantic Embedding Model,” *Advances in Neural Information Processing Systems*, 26(2013) 1~11.

Joachims, T., "Text categorization with support vector machines," *Technical report*, University of Dortmund, (1997).

Jolliffe, I.T., *Principal Component Analysis*, Springer-Verlag New York, Secaucus, NJ, (1989)

Kim, Y., "Convolutional neural networks for sentence classification," *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, 1746~1751.

Lee, M. and H. J. Lee, “Increasing Accuracy of Classifying Useful Reviews by Removing Neutral Terms,” Journal of Intelligent Information Systems, Vol.22, No.3(2016), 129~142.

Lee, M. and H. J. Lee, “Stock Price Prediction by Utilizing Category Neutral Terms: Text Mining Approach,” Journal of Intelligent Information Systems, Vol.23, No.2(2017), 123~138.

Lewis, D.D., "Naive (Bayes) at forty: The independence assumption in information retrieval," *Proceedings of ECML-98, 10th European Conference on Machine Learning,* (1998), 4~15.

Lewis, D.D., "Feature selection and feature extraction for text categorization," *Proceddings Speech and Natural Language Workshop, San Francisco*, (1992), 212~217.

Li, J., K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, and H. Liu, "Feature Selection: a data perspective," *ACM Computing Surveys(CSUR)*, Vol.50, No.6(2017), 94:1-94:45.

Landauer, T.K., P. W. Foltz, and D. Laham, "Introduction to Latent Semantic Analysis," *Discourse Processes*, Vol.25(1998), 259~84.

Mika, S., G. Ratsch, J. Weston, B. Scholkopf and K. -R. Muller, "Fisher discriminant analysis with kernels," *Proceedings, IEEE Workshop on Neural Network for Signal Processing*, (1999).

Mohan, P., I. Paramasivam, "A study on impact of dimensionality reduction on Naive Bayes classifier," *Indian Journal of Science and Technology*, Vol.10, No. 20(2017).

Peng, H., F. Long, C. Dong, "Feature selection based on mutual information: Criteria of max-dependence, max-relevance, min-redundancy",

IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.27, No.8(2005).

Pennington, J., R. Socher, and C. D. Manning. "Glove: Global vectors for word representation", *EMNLP*, (2014).

Peters, M., M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. "Deep contextualized word representations", *NAACL*, (2018).

Rapp, M., F.-J. Lübken, P. Hoffmann, R. Latteck, G. Baumgarten, and T. A. Blix, "PMSE dependence on aerosol charge, number density and aerosol size," *Journal of Geophysical Research*, Vol.108, No.D8(2003), 1~11.

Roweis, S.T. and Saul, L.K., "Nonlinear dimensionality reduction by Locally Linear Embedding," *Science,* Vol.290, No.5500(2000), 2323~2326.

Mika, S., G. Ratsch, J. Weston, B. Scholkopf, and K. -R Muller, "Fisher discriminant analysis with kernels," *Proceedings of IEEE Workshop on Neural Networks for Signal Processing*, (1999).

Sahami, M., "Learning limited dependence Bayesian classifiers". *Proceedings 2nd International Conference on Knowledge Discovery and Data Mining*, (1996), 334~338.

Sahlgren, M., "The distributional hypothesis," *Italian Journal of Linguistics*, Vol.20, No.1 (2008), 33~53.

Mikolov, T., K. Chen, G. Corrado, and Jeffrey Dean. "Efficient estimation of word representations in vector space", *ICLR Workshop*, (2013).

Yu, L.C., J. Wang, K. R. Lai, and X. Zhang, “Refining word embeddings for sentiment analysis”, *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, (2017), 545~550.

Zhang, R. and T. Tran, “An Information gain-based approach for recommending useful product reviews”, *Knowledge Information Systems*, Vol.26, No.3(2011), 419~434.

Zhou, P., W. Shi, J. Tian, Z. Qi, B. Li, H. Hao, and B. Xu. "Attention-based bidirectional long short-term memory networks for relation classification", *The 54th Annual Meeting of the Association for Computational Linguistics*, (2016), 207~213.

Zhu, L., G. Wang, and X. Zou, "Improved information gain feature selection method for Chinese text classification based on word embedding", *proceedings of the 6th International Conference on Software and Computer Applications*, (2017), 72~76.

Abstract

Selective Word Embedding for Sentence Classification by Considering Information Gain and Word Similarity

Min Seok Lee* · Seok Woo Yang** · Hong Joo Lee***

Dimensionality reduction is one of the methods to handle big data in text mining. For dimensionality reduction, we should consider the density of data, which has a significant influence on the performance of sentence classification. It requires lots of computations for data of higher dimensions. Eventually, it can cause lots of computational cost and overfitting in the model. Thus, the dimension reduction process is necessary to improve the performance of the model. Diverse methods have been proposed from only lessening the noise of data like misspelling or informal text to including semantic and syntactic information.

On top of it, the expression and selection of the text features have impacts on the performance of the classifier for sentence classification, which is one of the fields of Natural Language Processing. The common goal of dimension reduction is to find latent space that is representative of raw data from observation space. Existing methods utilize various algorithms for dimensionality reduction, such as feature extraction and feature selection. In addition to these algorithms, word embeddings, learning low-dimensional vector space representations of words, that can capture semantic and syntactic information from data are also utilized. For improving performance, recent studies have suggested methods that the word dictionary is modified according to the positive and negative score of pre-defined words.

The basic idea of this study is that similar words have similar vector representations. Once the feature selection algorithm selects the words that are not important, we thought the words that are similar to the selected words also have no impacts on sentence classification. This study proposes two ways to achieve more accurate classification that conduct selective word elimination under specific regulations and construct word embedding based on Word2Vec embedding. To select words having low importance from the text,

* Department of Business Administration, The Catholic University of Korea
** Department of Psychology, The Catholic University of Korea
*** Corresponding Author: Hong Joo Lee
Department of Business Administration, The Catholic University of Korea
43 Jibong-ro, Bucheon, Gyenggi 14662, Korea
Tel: +82-10-3887-1383, Fax: +82-2-2164-4280, E-mail: hongjoo@catholic.ac.kr

we use information gain algorithm to measure the importance and cosine similarity to search for similar words. First, we eliminate words that have comparatively low information gain values from the raw text and form word embedding. Second, we select words additionally that are similar to the words that have a low level of information gain values and make word embedding. In the end, these filtered text and word embedding apply to the deep learning models; Convolutional Neural Network and Attention-Based Bidirectional LSTM.

This study uses customer reviews on Kindle in Amazon.com, IMDB, and Yelp as datasets, and classify each data using the deep learning models. The reviews got more than five helpful votes, and the ratio of helpful votes was over 70% classified as helpful reviews. Also, Yelp only shows the number of helpful votes. We extracted 100,000 reviews which got more than five helpful votes using a random sampling method among 750,000 reviews. The minimal preprocessing was executed to each dataset, such as removing numbers and special characters from text data. To evaluate the proposed methods, we compared the performances of Word2Vec and GloVe word embeddings, which used all the words.

We showed that one of the proposed methods is better than the embeddings with all the words. By removing unimportant words, we can get better performance. However, if we removed too many words, it showed that the performance was lowered. For future research, it is required to consider diverse ways of preprocessing and the in-depth analysis for the co-occurrence of words to measure similarity values among words. Also, we only applied the proposed method with Word2Vec. Other embedding methods such as GloVe, fastText, ELMo can be applied with the proposed methods, and it is possible to identify the possible combinations between word embedding methods and elimination methods.

Key Words : Sentence Classification, Feature Selection, Information Gain, Word Similarity, Word Embedding

Received : June 23, 2019 Revised : September 22, 2019 Accepted : November 16, 2019
Publication Type : Regular Paper Corresponding Author : Hong Joo Lee

저 자 소 개

이 민 석
현재 가톨릭대학교 경영학과 학사과정 재학 중이다. 주요 관심 분야는 자연어 처리, 문장 분류, 감성 분석, 머신러닝, 딥러닝, 어텐션 등이다.

양 석 우
현재 가톨릭대학교 심리학과 학사과정 재학 중이다. 주요 관심 분야는 자연어 처리, 문장 분류, 감성 분석, 딥러닝 등이다.

이 홍 주
현재 가톨릭대학교 경영학전공 교수로 재직 중이다. KAIST 산업경영학과를 졸업하고 KAIST 테크노경영대학원에서 석사 및 박사학위를 취득하였다. 주요 관심분야는 데이터 분석, 지능형 정보시스템, 온라인 사용자들의 상호작용 등이다.