HANSHIN UNIVERSITY

Coursework Research Paper

Dept, of IT Transmedia Contents, Hanshin University

# 생성형 인공지능이 생성한 에세이 분류 모델의 모델 종류 및 벡터화 방식에 따른 분류 성능 비교

# Performance Comparison of Models that Classify Essays Generated by AI According to Model Types and Vectorization Methods

**Seungjae Moon**

Dept, of IT Transmedia Contents, Hanshin University

**[Abstract]**

This paper compares the performance of models that classify essays generated by artificial intelligence according to model types and vectorization methods. The experiment was conducted based on K-fold cross-validation, after dividing the case into TF-IDF vectorization and Word2Vec vectorization based on natural language process algorithm and creating a total of 4 models, Logistic Regression, Support Vector Machine, Random Forest, Voting Ansemble each case, and comparing various performance evaluation metrics, the performance of the Support Vector Machine model in the TF-IDF case came out highest.

**Key word :** AI generated essays, TF-IDF, Word2Vec, Classification, Natural language process, Machine learning

## Ⅰ. 서 론

인공지능의 발전으로 인한 인공지능 서비스의 확산과 함께 전문가뿐만 아니라 일반 대중들 또한 다양한 인공지능 서비스를 이용할 수 있게 되었다. 또한 이는 사회적으로 큰 영향을 미치고 있다. 특히, 현재 인공지능 분야의 뜨거운 감자인 Chat-GPT와 Midjounery와 같은 생성형 AI는 삶의 질을 높이는 데에 큰 역할을 하고 있다[1]. 하지만, 해당 인공지능 서비스들의 확산과 함께 큰 문제점이 발생하고 있다.

Chat-GPT와 같은 텍스트 생성 모델의 경우 생성된 자료의 신뢰성 문제가 뒤따르며 또 다른 문제점으로 윤리적 문제의 표절 문제가 발생하고 있다[2]. 연구자들이 높은 수준의 에세이와 논문 등을 생성하여 표절하거나, 학생들이 시험에서 챗봇 모델을 사용하여 부정행위를 하는 등 많은 사례가 등장하고 있다[3].

본 논문은 인공지능이 생성한 에세이와 실제 사람이 쓴 텍스트를 구별하기 위해 벡터화 방식과 머신러닝 모델에 따른 성능을 다양한 평가지표를 통해 비교 분석한다.

## Ⅱ. 관련 연구

자연어 처리에는 토큰화, 정규화 등 여러 가지 전처리 방식이 존재한다. 또한, 동일한 모델을 사용하여 자연어를 처리하더라도, 전처리 방식에 따라 속도와 정확도 등의 평가지표가 다양한 결과를 나타내기 때문에 자연어 처리에는 전처리 과정이 매우 중요한 과정이다[4]. 본 실험에서는 단어 및 벡터화 방식에 차이를 두어 실험을 진행한다.

### 2-1 TF-IDF

TF-IDF(Term Frequency-Inverse Document Frequency)란, 단어의 빈도수와 역 문서 빈도를 활용하여 DTM(Document-Term Matrix) 내 단어들마다 중요한 정도를 가중치로 주는 방식이다. 쉽게 말해, 단어의 빈도수를 통해 단어의 중요도를 알아내어 가중치를 부여하는 방식이다.

$$idf(t) = \log(\frac{n}{1+df(t)}) \quad (1)$$

TF-IDF의 tf(d,t) 는 특정 문서 d에서 특정 단어 t의 등장 횟수이다. df(t)는 특정 단어 t가 등장한 문서의 수이며, idf(t)는 df(t)에 반비례하는 수로, 위 수식을 따른다. 각 문서 내 특정 단어들의 TF 값과 IDF 값이 0 이상 1 이하의 값으로 표현되며, 이는 문서 내 빈도수를 이용하여 생성된 단어들의 가중치가 된다[5].

TF-IDF 벡터화 방식은 각 단어에 위와 같은 계산을 통해 생성된 TF와 IDF 값을 곱한 값을 인자로 취하여 문장을 벡터화한다.

### 2-2 Word2Vec

원-핫 벡터와 같은 희소 표현 방식은 단어 간 유사성을 표현할 수 없는 단점이 있다. 이를 해결하기 위해 '비슷한 문맥에서 **등장하는 단어들은 비슷한 의미를 가진다.'라는 가정을 따**라 단어 간 유사성을 고려하는 분산 표현 방식이 등장한다. 해당 방식을 이용하여 단어 간 유사성을 벡터화하는 워드임베딩이 등장한다. Word2Vec은 워드임베딩의 한 종류로, CBOW(Continous Bag of Words) 모델과 Skip-gram 모델이 존재한다[6].


**그림 1.** 슬라이딩 윈도우 방식(중심 단어와 주변단어)
**Fig. 1.** Sliding Windows Method(center and nearby words)

두 모델은 그림 1과 같은 Sliding Window 방식을 사용하여 주변 단어와 중심 단어를 예측한다.

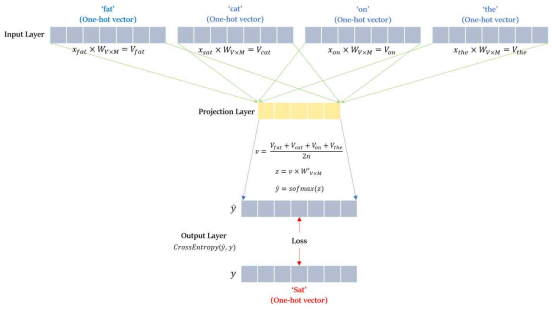


**그림 2.** CBOW(Continous Bag of Words) 모델 구조
**Fig. 2.** CBOW(Continous Bag of Words) model structure

CBOW 모델은 주변에 있는 단어들로부터 중간에 있는 단어를 예측하는 방식이다. 중심 단어를 벡터화할 때, 원-핫 인코딩한 주변 벡터들의 원-핫 벡터에 대해 임의의 가중치 W가 곱해지고, 은닉층에서 평균 벡터를 구하게 된다. 해당 평균 벡터는 두 번째 가중치 W′와 곱해지고, Softmax 함수를 취하여 스코어 벡터를 생성한다. 스코어 벡터는 중심 단어 원-핫 벡터의 값과 근사해지기 위해 Cross Entropy 손실함수를 사용하여 중심 단어의 원-핫 벡터를 출력한다.

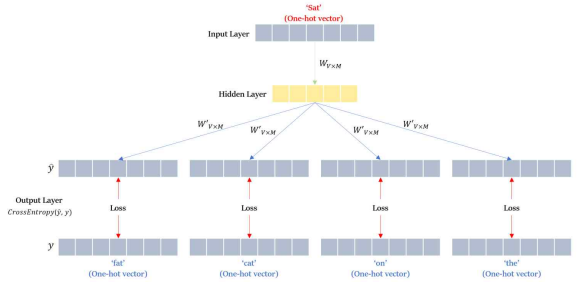


**그림 3.** Skip-gram 모델 구조
**Fig. 3.** Skip-gram model structure

Skip-gram 모델은 중간에 있는 단어를 통해 주변에 있는 단어를 예측하는 방식이다. 그림 3과 같이 CBOW 모델과 반대로, 원-핫 인코딩 된 중심 단어로부터 주변의 문맥 단어를 추측하여 원-핫 벡터화된 단어로 출력한다[7].

본 실험에서는 구글에서 제공하는 구글 뉴스의 말뭉치 3십억 개를 CBOW 모델로 사전 학습시킨 'GoogleNews-vectors-negative' Word2Vec를 사용한다.

## Ⅲ. 데이터

### 3-1 데이터 구성

데이터 세트는 Kaggle의 오픈소스 데이터 세트를 활용하였으며, 총 29,145개의 행과 2개의 열로 구성되어 있다. 2개의 열은 'text' 열과 'generated' 열로, 'text'열은 텍스트, 'generated'열은 각 텍스트에 대한 레이블을 뜻한다. 해당 데이터 세트는 사람이 쓴 텍스트를 0, 인공지능이 생성한 텍스트를 1로 레이블링 되어있다.

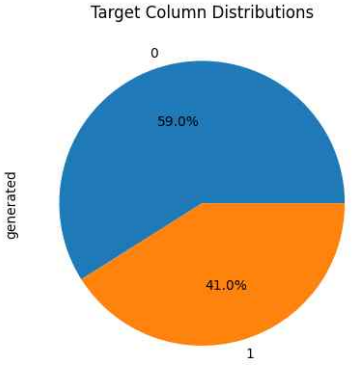


**그림 4.** 클래스 별 데이터 개수 비율
**Fig. 4.** Data count ratio by class

데이터의 중복 여부를 확인한 결과 중복된 데이터의 존재로 인해 중복된 데이터를 제거하여 27,340개의 데이터로 실험을 진행한다.

### 3-2 데이터 전처리

실험을 진행하기 전, 자연어 데이터는 모델의 입력값으로 사용하기 위한 전처리 과정이 필수적이며, 각 방식에 필요한 형태로 전처리하는 과정을 진행한다.

각 텍스트 데이터에서 느낌표(!), 콤마(,)와 같은 특수문자를 제거하고, python nltk 패키지의 'stopword'에 등록된 영어 불용어를 제거한다. 특수문자와 불용어가 제거된 텍스트는 단어 단위로 토큰화한다. 토큰화된 데이터는 마지막으로 벡터화를 거쳐 모델의 입력값으로 사용할 수 있다. 본 실험에서는 각 머신 러닝 모델에 TF-IDF와 Word2Vec의 서로 다른 방식으로 벡터화한 데이터를 입력값으로 사용하여 비교한다.

일반적으로 무작위로 분리한 학습데이터와 테스트 데이터로 학습한 모델과 결과는 과적합과 결과에 대한 불확실성이 생길 수 있다. 따라서, K-fold 교차 검증 방식을 사용하여 학습데이터와 테스트 데이터를 분리한다. 또한, 그림 1과 같이 데이터의 분포가 59:41로 불균형하므로, 학습데이터와 테스트 데이터가 가지는 레이블 분포도가 유사하도록 데이터를 분리하여 검증하는 Stratified K-fold 교차 검증을 사용한다.

## Ⅳ. 실 험

### 4-1 실험 환경 및 방법

본 실험은 Google Colab TPU 환경에서 진행하였으며, Stratified K-fold cross validation (K=5)를 사용하여 학습 데이터와 테스트 데이터를 한 모델당 총 5번의 실험을 진행하였다. 모델은 Logistic Regression, SVC(Support Vector Classification), Random Forest와 앞의 세 모델을 Voting 앙상블 기법으로 생성한 모델로, 총 4개의 모델을 TF-IDF 벡터화한 데이터와 Word2Vec으로 벡터화한 데이터를 입력값으로 하는 두 가지 케이스로 나누어 비교한다.

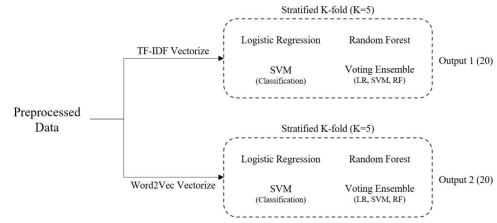


**그림 5. 실험 워크플로우**
**Fig. 5.** Experiment workflow

### 4-2 실험 결과 및 분석

본 실험에서는 Logistic Regression, SVM, Random Forest, Voting 등 서로 다른 4개의 모델에 다른 방식의 벡터화를 진행한 21,872개의 텍스트 데이터를 K-fold를 이용하여 학습한 모델을 생성하고 평가하였다.

| TF-IDF | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Logistic Regression | | SVM | | Random Forest | | Voting | |
| **Fold-1** | | | | | | | | |
| | **Person** | **AI** | **Person** | **AI** | **Person** | **AI** | **Person** | **AI** |
| Person | 3146 | 78 | 3189 | 35 | 3175 | 49 | 3176 | 48 |
| AI | 95 | 2149 | 38 | 2206 | 107 | 2137 | 61 | 2183 |
| precision | 0.97 | 0.96 | 0.99 | 0.98 | 0.97 | 0.98 | 0.98 | 0.98 |
| recall | 0.98 | 0.96 | 0.99 | 0.98 | 0.98 | 0.95 | 0.99 | 0.97 |
| f1-score | 0.97 | 0.96 | 0.99 | 0.98 | 0.98 | 0.96 | 0.98 | 0.98 |
| accuracy | 0.97 | | 0.99 | | 0.97 | | 0.98 | |
| **Fold-2** | | | | | | | | |
| Person | 3130 | 94 | 3199 | 25 | 3178 | 46 | 3182 | 42 |
| AI | 94 | 2150 | 36 | 2208 | 127 | 2117 | 70 | 2174 |
| precision | 0.97 | 0.96 | 0.99 | 0.99 | 0.96 | 0.98 | 0.98 | 0.98 |
| recall | 0.97 | 0.96 | 0.99 | 0.98 | 0.99 | 0.94 | 0.99 | 0.97 |
| f1-score | 0.97 | 0.96 | 0.99 | 0.99 | 0.97 | 0.96 | 0.98 | 0.97 |
| accuracy | 0.97 | | 0.99 | | 0.97 | | 0.98 | |
| **Fold-3** | | | | | | | | |
| Person | 3128 | 96 | 3197 | 27 | 3182 | 42 | 3185 | 39 |
| AI | 93 | 2151 | 37 | 2207 | 127 | 2117 | 73 | 2171 |
| precision | 0.97 | 0.96 | 0.99 | 0.99 | 0.96 | 0.98 | 0.98 | 0.98 |
| recall | 0.97 | 0.96 | 0.99 | 0.98 | 0.99 | 0.94 | 0.99 | 0.97 |
| f1-score | 0.97 | 0.96 | 0.99 | 0.99 | 0.97 | 0.96 | 0.98 | 0.97 |
| accuracy | 0.97 | | 0.99 | | 0.97 | | 0.98 | |
| **Fold-4** | | | | | | | | |
| Person | 3134 | 91 | 3189 | 36 | 3176 | 49 | 3182 | 43 |
| AI | 98 | 2147 | 29 | 2214 | 109 | 2134 | 65 | 2178 |
| precision | 0.97 | 0.96 | 0.99 | 0.98 | 0.97 | 0.98 | 0.98 | 0.98 |
| recall | 0.97 | 0.96 | 0.99 | 0.99 | 0.98 | 0.95 | 0.99 | 0.97 |
| f1-score | 0.97 | 0.96 | 0.99 | 0.99 | 0.98 | 0.96 | 0.98 | 0.98 |
| accuracy | 0.97 | | 0.99 | | 0.97 | | 0.98 | |
| **Fold-5** | | | | | | | | |
| Person | 3118 | 107 | 3191 | 34 | 3163 | 62 | 3180 | 45 |
| AI | 86 | 2157 | 44 | 2199 | 114 | 2129 | 69 | 2174 |
| precision | 0.98 | 0.98 | 0.99 | 0.99 | 0.97 | 0.97 | 0.98 | 0.98 |
| recall | 0.99 | 0.97 | 0.98 | 0.98 | 0.98 | 0.95 | 0.99 | 0.97 |
| f1-score | 0.98 | 0.98 | 0.99 | 0.98 | 0.97 | 0.96 | 0.98 | 0.97 |
| accuracy | 0.98 | | 0.99 | | 0.97 | | 0.98 | |
| **K-Fold Mean** | | | | | | | | |
| precision | 0.972 | 0.964 | 0.990 | 0.986 | 0.966 | 0.978 | 0.980 | 0.980 |
| recall | 0.976 | 0.962 | 0.988 | 0.972 | 0.984 | 0.946 | 0.990 | 0.970 |
| f1-score | 0.972 | 0.964 | 0.990 | 0.986 | 0.974 | 0.960 | 0.980 | 0.974 |
| accuracy | 0.972 | | 0.990 | | 0.970 | | 0.980 | |

**표 1.** TF-IDF 벡터화한 데이터를 입력값으로 한 분류 모델의 혼동행렬 및 성능평가지표
**Table. 1.** Confusion matrix and evaluation metrics of classification models used TF-IDF vectorized data as input

표 1은 TF-IDF 벡터화한 데이터를 학습한 모델들의 성능평가 결과이다. 모든 시행에서 각 모델의 혼동행렬을 확인한 결과, Random Forest 모델은 AI가 생성한 텍스트를 사람이 쓴 텍스트로 예측하는 오분류가 모든 시행에서 100개 이상으로, 다른 모델이 비해 높은 오분류율을 보였다. 4개의 모델 중 전체 시행에서 가장 좋은 정확도를 보인 모델은 SVC였다. 정밀도, 재현율, f1-score 평가지표의 평균이 0.99 이상이며 평균 0.99의 매우 높은 정확도를 보였다.

| Word2Vec | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Logistic Regression | | SVM | | Random Forest | | Voting | |
| **Fold 1** | | | | | | | | |
| | **Person** | **AI** | **Person** | **AI** | **Person** | **AI** | **Person** | **AI** |
| Person | 3142 | 82 | 3195 | 29 | 3170 | 54 | 3171 | 53 |
| AI | 96 | 2148 | 60 | 2184 | 84 | 2160 | 76 | 2168 |
| precision | 0.97 | 0.96 | 0.98 | 0.99 | 0.97 | 0.98 | 0.98 | 0.98 |
| recall | 0.97 | 0.96 | 0.99 | 0.97 | 0.98 | 0.96 | 0.98 | 0.97 |
| f1-score | 0.97 | 0.96 | 0.99 | 0.98 | 0.98 | 0.97 | 0.98 | 0.97 |
| accuracy | 0.97 | | 0.98 | | 0.97 | | 0.98 | |
| **Fold 2** | | | | | | | | |
| Person | 3144 | 80 | 3208 | 16 | 3188 | 36 | 3190 | 34 |
| AI | 93 | 2151 | 55 | 2189 | 73 | 2171 | 77 | 2167 |
| precision | 0.97 | 0.96 | 0.98 | 0.99 | 0.98 | 0.98 | 0.98 | 0.98 |
| recall | 0.98 | 0.96 | 1.00 | 0.98 | 0.99 | 0.97 | 0.99 | 0.97 |
| f1-score | 0.97 | 0.96 | 0.99 | 0.98 | 0.98 | 0.98 | 0.98 | 0.98 |
| accuracy | 0.97 | | 0.99 | | 0.98 | | 0.98 | |
| **Fold 3** | | | | | | | | |
| Person | 3143 | 81 | 3208 | 16 | 3184 | 40 | 3181 | 43 |
| AI | 96 | 2148 | 55 | 2189 | 85 | 2159 | 69 | 2175 |
| precision | 0.97 | 0.96 | 0.98 | 0.99 | 0.97 | 0.98 | 0.98 | 0.98 |
| recall | 0.97 | 0.96 | 1.00 | 0.98 | 0.99 | 0.96 | 0.99 | 0.97 |
| f1-score | 0.97 | 0.96 | 0.99 | 0.98 | 0.98 | 0.97 | 0.98 | 0.97 |
| accuracy | 0.97 | | 0.99 | | 0.98 | | 0.98 | |
| **Fold 4** | | | | | | | | |
| Person | 3138 | 87 | 3203 | 22 | 3177 | 48 | 3176 | 49 |
| AI | 72 | 2171 | 46 | 2197 | 68 | 2175 | 60 | 2183 |
| precision | 0.98 | 0.96 | 0.99 | 0.99 | 0.98 | 0.98 | 0.98 | 0.98 |
| recall | 0.97 | 0.97 | 0.99 | 0.98 | 0.99 | 0.97 | 0.98 | 0.97 |
| f1-score | 0.98 | 0.96 | 0.99 | 0.98 | 0.98 | 0.97 | 0.98 | 0.98 |
| accuracy | 0.97 | | 0.99 | | 0.98 | | 0.98 | |
| **Fold 5** | | | | | | | | |
| Person | 3132 | 93 | 3198 | 27 | 3168 | 57 | 3170 | 55 |
| AI | 83 | 2160 | 54 | 2189 | 71 | 2172 | 64 | 2179 |
| precision | 0.97 | 0.96 | 0.98 | 0.99 | 0.98 | 0.97 | 0.98 | 0.98 |
| recall | 0.97 | 0.96 | 0.99 | 0.98 | 0.98 | 0.97 | 0.98 | 0.97 |
| f1-score | 0.97 | 0.96 | 0.99 | 0.98 | 0.98 | 0.97 | 0.98 | 0.97 |
| accuracy | 0.97 | | 0.99 | | 0.98 | | 0.98 | |
| **K-Fold Mean** | | | | | | | | |
| precision | 0.972 | 0.960 | 0.982 | 0.990 | 0.976 | 0.978 | 0.980 | 0.980 |
| recall | 0.972 | 0.962 | 0.994 | 0.978 | 0.986 | 0.970 | 0.984 | 0.970 |
| f1-score | 0.972 | 0.960 | 0.990 | 0.980 | 0.980 | 0.972 | 0.980 | 0.974 |
| accuracy | 0.970 | | 0.988 | | 0.978 | | 0.980 | |

**그림 2.** Word2Vec로 벡터화한 데이터를 입력값으로 한 분류 모델의 혼동행렬 및 성능평가지표
**Table. 2.** Confusion matrix and evaluation metrics of classification models used vectorized data by Word2Vec as input

표 2는 Word2Vec 모델을 이용하여 벡터화한 데이터를 학습한 모델들의 성능평가 결과이다. Logistic Regression의 경우 모든 시행에서 성능이 다른 모델에 비해 상대적으로 낮으며, 최종 평균 성능 또한 낮은 것을 확인할 수 있다. 시행에서 오분류한 개수가 100개 미만이었으며 전체적인 성능평가가 매우 높게 나왔다.

TF-IDF 벡터화와 Word2Vec을 이용한 벡터화 두 케이스의 성능평가 지표를 모두 확인한 결과, 두 모델 모두 Logistic Regression 모델과 Random Forest 모델이 SVM과 Voting 앙상블 방식의 모델에 비해 낮은 분류 성능을 띄는 것을 확인할 수 있다. 각 케이스의 두 모델을 비교하였을 때, AI 레이블에 대한 재현율은 Word2Vec의 SVM 모델이 가장 높았다. 또한, 전체 케이스의 모든 모델 중 정확도가 0.99로 가장 높은 정확도를 띄는 것을 확인할 수 있다.

## Ⅴ. 결 론

본 실험에서는 벡터화 방식과 모델에 따라 인공지능이 생성한 텍스트를 분류하는 모델의 성능을 평가하였다. 모든 모델이 0.9 이상의 매우 높은 성능을 띄었고, Word2Vec을 이용하여 벡터화한 데이터를 토대로 생성된 Support Vector Machine이 다양한 성능평가 지표에서 가장 높은 성능을 보였다. Word2Vec의 특성상 수십억 개의 말뭉치를 사전 학습한 모델을 토대로 학습 데이터를 벡터화할 수 있었으며, 고차원 공간에서도 뛰어난 성능을 보이는 SVM 모델을 통해 더욱 높은 성능을 보였음을 추측할 수 있다. 해당 실험을 통해 약 99%로 인공지능이 생성한 텍스트를 판별하는 모델을 생성할 수 있었다. 하지만, 본 실험에 쓰인 텍스트 데이터에서는 각 방식은 모두 매우 높은 성능을 보였지만, 다른 데이터 세트에서도 성능을 검증할 필요성이 있다. 또한, 현재 생성형 모델이 빠르게 발전하고 있는 만큼, 인공지능은 더욱 정교하고 섬세한 텍스트를 생성하기 때문에 생성된 텍스트를 판별하는 모델에 대한 지속적인 발전이 필요하고, 다양한 방식으로 시도하는 것이 중요하며, 이는 생성형 인공지능의 성장과 함께 상호 발전할 것으로 기대된다.

## References


[1] M Jovanovic, M Campbell, “Generative artificial intelligence: Trends and prospects”, *IEEE Ccomputer society*, Vol.55, No.10, pp. 107-102, Oct. 2022.

[2] George Lawton, “Generative AI ethics: 8 biggest concerns and risks”, *TechTarget*, Nov 2023.

[3] Yogesh K. Dwivedi et al., “Opinion Paper: ‘So what if ChatGPT wrote it?’ Multidisciplinary perspectives on opportunities, challenges and implications of generative conversational AI for research, practice and policy”, *International Journal of Information Management*, Vol.71, Aug. 2023.

[4] Christine P. Chai, “Comparison of text preprocessing methods” *Edubirdie*, Vol.29, No.3, pp. 509-553, May. 2023.

[5] C. -z. Liu, Y. -x. Sheng, Z. -q. Wei and Y. -Q. Yang, "Research of Text Classification Based on Improved TF-IDF Algorithm," *2018 IEEE International Conference of Intelligent Robotic and Control Engineering (IRCE)*, pp. 218-222, Oct. 2018.

[6] Rajvi Shah, “Word2Vec”, *Cambridge University Press*, Vol.23, No.1, pp. 155-162, Dec. 2016.

[7] Zeyu Xiong, Qiangqiang Shen, Yueshan Xiong, Yijie Wang, Weizi Li, “New Generation Model of Word Vector Representation Based on CBOW or Skip-Gram.”, *CMC-Comput Mater Con*, Vol.58, No.1, pp.259-273, 2019