# Bag of Characters를 응용한 단어의 벡터 표현 생성 방법*

이찬희† · 이설화† · 임희석†
† 고려대학교 정보대학 컴퓨터학과

# Word Vectorization Method Based on Bag of Characters

Chanhee Lee† · Seolhwa Lee† · Heuiseok Lim†
† Dept. of Computer Science and Engineering, College of Informatics, Korea University

## 요 약

인공 신경망 기반 자연어 처리 시스템들에서 단어를 벡터로 변환할 때, 크게 색인 및 순람표를 이용하는 방법과 합성곱 신경망이나 회귀 신경망을 이용하는 방법이 있다. 이 때, 전자의 방법을 사용하려면 시스템이 수용 가능한 어휘집이 정의되어 있어야 하며 새로운 단어를 어휘집에 추가하기 어렵다. 반면 후자의 방법을 사용하면 단어를 구성하는 문자들을 바탕으로 벡터 표현을 생성하기 때문에 어휘집이 필요하지 않지만, 추가적인 인공 신경망 구조가 필요하기 때문에 모델의 복잡도와 파라미터의 수가 증가한다는 단점이 있다. 본 연구에서는 위 두 방법의 한계를 극복하고자 Bag of Characters를 응용하여 단어를 구성하는 문자들의 집합을 바탕으로 벡터 표현을 생성하는 방법을 제안한다. 제안된 방법은 문자를 기반으로 동작하기 때문에 어휘집을 정의할 필요가 없으며, 인공 신경망 구조가 사용되지 않기 때문에 시스템의 복잡도도 증가시키지 않는다. 또한, 단어의 벡터 표현에 단어를 구성하는 문자들의 정보가 반영되기 때문에 Out-Of-Vocabulary 단어에 대한 성능도 어휘집을 사용하는 방법보다 우수할 것으로 기대된다.

## 1. 서 론

최근 단어를 입력으로 받는 인공 신경망 기반 자연어 처리 시스템들(예를 들어, 품사 태깅, 워드 임베딩, 개체명 인식 등)은 대부분 단어를 서로 독립적인 단위로 취급하며, one-hot 인코딩이나 순람표를 사용하여 이를 벡터로 변환한다[1][2]. 하지만 이러한 방법을 사용하려면 시스템이 사용 가능한 한정된 단어들의 목록인 어휘집이 정의되어 있어야 하며, 새로운 단어를 어휘집에 추가하기 어렵다. 또한, 이러한 시스템은 파라미터의 수가 어휘집의 크기와 비례하여 증가한다는 단점이 있다. 어휘집에 포함되지 못한 (Out-Of-Vocabulary, OOV) 단어에 대한 처리도 문제가 된다. 여기에 가장 보편적으로 사용되는 방법은 모든 OOV 단어들을 이를 나타내는 하나의 특수한 기호로 치환하는 것인데, 이는 OOV 단어에 대한 시스템의 정확도를 크게 하락시키고 자연스럽게 시스템의 전반적인 성능에도 악영향을 미치게 된다. 합성곱 신경망[3][4]이나 회귀 신경망[5]을 사용하여 단어가 아닌 문자 단위로 동작하는 방법들도 시도되었지만, 이러한 방법은 추가적인 신경망 구조가 필요하여 시스템의 복잡도가 증가하게 된다.

본 연구에서는 이러한 기존의 단어 처리 방법의 한계를 극복하기 위하여 Bag of Characters 방법에 착안한 새로운 단어 처리 방법을 제안한다. 제안된 방법은 단어를 구성하는 문자 정보를 바탕으로 단어 벡터를 생성하기 때문에 한정된 어휘집이 필요하지 않으며, 합성곱 신경망이나 회귀 신경망과 같은 추가적인 구조를 사용하지 않으므로 시스템의 복잡도도 증가하지 않는다. 또한, 학습 데이터의 어휘집에 존재하지 않았던 단어도 동일한 기호로 치환되는 대신 단어를 구성하는 문자 정보를 바탕으로 한 고유의 벡터로 변환되기 때문에 OOV 단어에 대한 시스템의 정확도 향상도 기대할 수 있다.

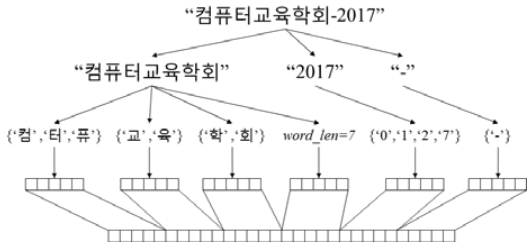
## 2. 연구 방법

### 2.1 단어 벡터 생성 방법

단어를 이를 구성하는 문자들을 바탕으로 벡터로 변환할 때 가장 큰 문제가 되는 것은 단어의 길이가 가변적이라는 점이다. 단순히 각 문자를 벡터로 변환하고 이들을 결합하게 되면 문자의 수에 따라 단어 벡터

* 이 논문은 2017년도 정부(산업통상자원부)의 재원으로 한국산업기술평가관리원의 지원을 받아 수행된 연구임(No. 10079423).

의 길이가 변하게 되는데, Feed Forward Neural Network은 입력 벡터의 크기가 고정되어 있으므로 이러한 방법은 사용할 수 없다. 합성곱 신경망이나 회귀




[그림 1] 제안된 방법을 이용하여 "컴퓨터교육학회-2017"에 대한 벡터 표현을 생성하는 과정

신경망을 문자에 적용하면 단어의 길이와 독립적인 일정한 길이의 벡터를 얻을 수 있지만, 앞서 언급한대로 시스템의 복잡도와 파라미터의 수가 증가한다는 단점이 있다.

Bag of Characters는 단어를 문자의 집합으로 나타내는 방법이다. 한 언어를 구성하는 문자의 수는 단어의 수보다 매우 적고 크게 변하지 않으므로, Bag of Characters를 이용하면 단어를 일정한 크기의 벡터로 변환할 수 있다. 하지만 이를 자연어 처리 시스템의 단어 벡터 표현 방법으로 사용할 경우 그 과정에서 문자 순서에 대한 정보가 소실되므로, '가나-나가'나 '국왕-왕국'과 같이 구성하는 문자는 동일하지만 순서만 다른 단어들이 동일한 벡터로 표현되는 '단어 충돌' 현상이 발생한다. 이는 시스템의 성능 하락으로 이어질 수 있으므로 단어 충돌 횟수를 줄이는 것이 중요하다.

본 연구에서는 Bag of Character가 가지는 이러한 문제를 극복하고자 단어를 여러 조각으로 분할하여 벡터로 변환하는 방법을 제안한다. 구체적인 방법은 다음과 같다. 우선 단어를 한글, 숫자, 특수문자로 분리한다. 각 단어 조각은 아래 기술된 방법에 따라 Bag of Characters를 응용한 방법을 이용하여 벡터로 변환된다. 여기에 추가적으로 단어의 길이를 one-hot encoding을 이용하여 벡터로 표현한다. 마지막으로, 이렇게 얻어진 벡터들을 결합하여 최종적으로 단어에 대한 벡터 표현을 형성한다. 이 과정을 도식화하면 [그림 1]과 같다. 각 단어 조각을 벡터로 변환하는 방법은 아래와 같다.

#### 2.1.1 한글 문자열의 벡터화

단어 충돌 횟수를 줄이기 위해 한글 문자열 부분을 더 작은 조각으로 분할한다. 이 때, 각 조각에 속한 글자의 수가 최대한 균일하도록 분할을 수행한다. 정확한 분할 위치는 아래의 식(1)에 따라 계산된다.

$$p_i = \left\lfloor \frac{l(w) \times i}{n} \right\rfloor \quad (1)$$

$p_i$는 $i$번째 조각의 분할 위치, $w$는 한글 문자열, $l(w)$는 문자열 $w$의 길이이며, $n$은 분할 후 조각의 개수이다. 문자열을 몇 부분으로 분할할지 결정하는 $n$은 하이퍼 파라미터로, 실험 결과를 바탕으로 최적의 값을 선택한다.

분할 후 각 문자열은 Bag of Characters 표현 방법을 사용하여 벡터로 변환한다. 여기에 추가적으로 문자 중복 여부를 표시하는 비트를 추가하는데, 각 문자열 조각에 같은 문자가 2회 이상 반복될 경우 1, 그 외의 경우는 0의 값을 갖는다. 마지막으로, 이렇게 만들어진 벡터들을 결합하여 한글 문자열의 벡터 표현을 얻는다.

#### 2.1.2 숫자 문자열의 벡터화

숫자 문자열은 단순히 Bag of Characters 표현 방법을 사용하여 벡터로 변환하며, 여기에 중복 여부를 표시하는 비트를 추가하여 숫자 문자열의 벡터 표현을 생성한다.

#### 2.1.3 특수 문자 문자열의 벡터화

시스템 대상 언어의 말뭉치에서의 문자 등장 빈도를 바탕으로 가장 빈도가 높은 특수 문자 N개를 선정한 후, 이를 Bag of Characters를 이용하여 벡터로 표현한다. 여기에 추가로 N개에 속하지 못한 특수 문자들을 '기타' 문자로 분류하여 벡터 표현에 추가한다.

#### 2.1.4 단어 길이 정보의 벡터화

단어 충돌 빈도를 더 감소시키기 위하여, 단어의 길이를 벡터로 표현하여 최종적인 벡터 표현에 추가한다. 0부터 20까지의 숫자와 이보다 긴 단어를 표현하기 위한 위치를 포함하여 총 21 비트 길이의 벡터를 이에 사용한다.

## 3. 실험 설계

본 연구에서 제안된 단어의 벡터 표현 방법의 우수성을 검증하기 위해서 자연어 처리 시스템을 구현하여 성능을 기존 모델과 비교해볼 수 있다. 회귀 신경망을 이용한 품사 태깅[3][6][7] 시스템은 자연어 처리 시스템의 전처리 단계로 많이 활용되며, 단어를 입력으로 사용하기 때문에 제안된 방법의 성능을 검증하기 적절할 것으로 예상된다. 비교 모델로는 색인 및 순람표를 이용하여 단어를 벡터로 표현하는 모델과 문자 단위로 합성곱 신경망이나 회귀 신경망을 사용하는 모델을 사용할 수 있다. 제안된 모델은 전자의 모델과 대비하여서는 정확도가, 후자의 모델과 대비하여서는 학습 및 실행 속도가 우수할 것으로 예상되므로, 제안된 방법

을 사용한 모델 및 모든 비교 대상 모델을 대상으로 정확도, 학습 속도, 실행 속도, 학습 파라미터의 수를 측정 및 비교함으로써 본 방법의 우수성을 정량적으로 검증할 수 있다.

## 4. 결론 및 향후 연구

본 연구에서는 Bag of Characters를 응용하여 단어를 벡터로 변환하는 단순하면서도 효과적인 방법을 제안하였다. 제안된 모델은 단어를 구성하는 문자들의 집합을 바탕으로 동작하기 때문에 어휘집이 정의되어야 할 필요가 없다. 또한, 기존의 합성곱 신경망이나 회귀 신경망을 이용하여 단어를 구성하는 문자열을 바탕으로 벡터를 생성하는 방법과 달리 추가적인 인공 신경망 구조가 필요하지 않기 때문에 시스템의 복잡도나 파라미터의 수가 더 낮다.

향후 연구로는 제안된 방법을 구현하여 품사 태깅이나 개체명 인식과 같은 시스템에 적용한 후 기존의 방법들과 정량적인 비교를 수행하여야 한다. 추가적으로, 제안된 방법의 가장 큰 한계점인 단어 충돌 현상의 빈도를 더욱 감소시키기 위해 추가적인 단어 정보들을 벡터 표현에 추가하는 방법을 연구할 수 있다.

## 참 고 문 헌


[1] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems (pp. 3111-3119).

[2] Huang, Z., Xu, W., & Yu, K. (2015). Bidirectional LSTM-CRF models for sequence tagging. arXiv preprint arXiv:1508.01991.

[3] Santos, C. D., & Zadrozny, B. (2014). Learning character-level representations for part-of-speech tagging. In Proceedings of the 31st International Conference on Machine Learning (ICML-14) (pp. 1818-1826).

[4] Kim, Y., Jernite, Y., Sontag, D., & Rush, A. M. (2016, February). Character-Aware Neural Language Models. In AAAI (pp. 2741-2749).

[5] Ling, W., Lu?s, T., Marujo, L., Astudillo, R. F., Amir, S., Dyer, C., ... & Trancoso, I. (2015). Finding function in form: Compositional character models for open vocabulary word representation. arXiv preprint arXiv:1508.02096.

[6] Haji?, J., Raab, J., & Spousta, M. (2009, March). Semi-supervised training for the averaged perceptron POS tagger. In Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics (pp. 763-771). Association for Computational Linguistics.

[7] Manning, C. D. (2011, February). Part-of-speech tagging from 97% to 100%: is it time for some linguistics?. In International Conference on Intelligent Text Processing and Computational Linguistics (pp. 171-189). Springer, Berlin, Heidelberg.