속보 문서 분류를 위한 RNN 기반의 단어 임베딩

김지영, 류성원, 이형욱, 선지민, 조성준

[a] *Department of Industrial Engineering and Institute for Industrial Systems Innovation, Seoul National University*
1, Gwanak-ro, Gwanak-gu, Seoul, 08826, South Korea
Tel: +82-2-880-7025, E-mail: jeeyung.kim@dm.snu.ac.kr
Tel: +82-2-880-7025, E-mail: lyusungwon@dm.snu.ac.kr
Tel: +82-2-880-7025, E-mail: hyeonguk21*@dm.snu.ac.kr*
Tel: +82-2-880-7025, E-mail: jiminsun@dm.snu.ac.kr
Tel: +82-2-880-7025, E-mail: zoon@dm.snu.ac.kr

[b] *서울대학교 산업공학과 및 산업시스템혁신연구소*
08826, 서울특별시 관악구 관악로 1

Abstract

단어를 분산 표현하는 것은 Natural Language Processing(NLP) 분야의 다른 문제들을 해결하기 위해 선행되어야 하는 연구 분야이다. 기존의 단어 임베딩 방법은 학습 데이터의 말뭉치에 없는 단어의 임베딩은 얻을 수 없는 Out of Vocabulary(OOV) 문제점을 갖고 있다. 이를 해결하고자, 본 논문에서는 RNN(Recurrent Neural Network) based Skip-gram을 사용한 단어 임베딩 생성 모델을 제안한다. 단어를 글자 단위로 RNN 모듈에 입력시켜 얻은 단어 임베딩을 skip-gram 모듈에서 중심, 주변 단어를 표현하는 데 활용한다 결과적으로, RNN 모듈에서는 단어의 형태학적 특징을 잡아내고, Skip-gram 방식으로 학습을 진행하여 단어의 의미론적 및 구문론적 특징을 학습하게 된다. 또한 학습의 결과로 모델의 RNN 모듈에 해당하는 단어 임베딩 생성기를 얻게 되므로, 학습 말뭉치에 없었던 새로운 단어의 임베딩 생성도 가능할 것이라 기대한다.

더불어, 단어 임베딩 생성기에 Recurrent Neural Network 기반의 분류기를 추가하면 문서 분류 문제 해결이 가능하다. 임베딩 생성기를 이용하면, 기존의 단어 임베딩 방법이나 bag-of-words 방법과는 달리 학습 말뭉치에 없던 단어를 포함한 문서의 분류가 가능할 것이다. 속보 문서 분류에 응용하여, 국가 안보에 도움이 될 수 있다.

Keywords: 문서 분류, 단어 분산 표현, Recurrent Neural Network, Skip-gram, OOV

Security; Text Classification; Natural Language Processing

서론

단어를 분산 표현하는 것은 Natural Language Processing(NLP) 분야의 주요한 문제인 QA(Question Answering), MT(Machine Translation), SRL(Semantic Role Labeling) 등을 해결하기 위해 선행되어야 하는 연구이다. 효과적으로 단어를 분산 표현하는 연구는 과거부터 활발히 진행되었으며, 대표적인 단어 임베딩 방법에는 Skip-gram(2013), Cbow(2013), Glove(2014), FastText(2016)등 이 있다.

하지만 기존의 단어 분산 표현 방법들은 공통의 문제점을 가지고 있다. 학습 데이터의 말뭉치에 없는 단어의 임베딩은 얻을 수 없는 Out of Vocabulary(OOV)문제가 존재한다. 학습한 단어에 없는 새로운 단어가 등장했을 경우에는 이를 랜덤하게 초기화하거나 특정한 값을 가진 벡터로 초기화하지만, 새로운 단어가 많은 도메인이나 정밀한 단어 분산 표현이 중요한 특정 문제에서는 보다 정확한 단어 임베딩이 필요하다. 또한, 기존의 단어 표현을 위해서는 stemming, lemmatization 등의 텍스트 데이터 전처리 과정이 필요하며, 이는 시간과 자원이 매우 많이 소요된다는 문제점이 있다.

본 논문에서는 기존 모델의 문제점을 해결하고자, 새로운 단어 분산 표현 학습 방법을 제안한다. RNN(Recurrent Neural Network)와 Skip-gram 을 결합한 단어 임베딩 생성 모델이다. 구체적으로, 분산 표현이 필요한 단어를 글자 단위로 RNN 모듈에 입력시켜 마지막 hidden state를 단어 임베딩으로 활용한다. Skip-gram(2013) 의 학습 방법과 유사하게, 중심 단어를 입력시켰을 때 주변 단어가 나올 확률을 높이는 방향으로 학습시키는데, 이때 RNN의 결과물인 임베딩을 활용하여 중심 단어와, 주변 단어들을 표현한다. 결과적으로, RNN 모듈에서는 단어의 형태학적 특징, 즉 접두사 및 시제 등을 잡아내고, Skip-

gram 방식으로 학습을 진행하며 단어의 문맥상의 의미를 학습한다.

본 논문에서 제안하는 모델의 학습 결과로, 모델의 RNN 모듈에 해당하는 단어 임베딩 생성기를 얻게 된다. 기존의 모델과는 달리 단어의 구문론적, 의미론적 특성뿐만 아니라 단어 자체의 형태적 특징을 고려하여 임베딩 하였기 때문에 더 정밀한 임베딩 결과를 얻을 것이라 기대한다. 또한, 학습의 결과로 생성기를 얻기 때문에 새로운 단어(OOV)가 왔을 때, 기존에 아무 정보를 담지 못하는 초기화 방법과는 달리 보다 효과적으로 단어를 표현할 수 있다. 더불어, 이러한 접근은 새로운 단어가 생길 때마다 새롭게 전체 단어 임베딩을 학습할 필요가 없기 때문에 시간과 자원을 아낄 수 있다는 장점도 갖는다. 결론적으로, 본 논문에서 제안한 임베딩 방법을 활용하면 위에 언급했던 NLP의 주요 문제들을 푸는 데에 기여할 수 있을 것이라 기대한다.

NLP 주요 문제들 중, 임베딩 생성기를 활용하여 문서 분류 문제를 풀 수 있다. 문서 분류기를 통해 주어진 속보 문서의 소재를 빠르게 파악할 수 있다. 국가 안보를 위해 수많은 문서를 빠르게 분류해야 하는 경우에 응용될 수 있는 기술이다.

제안한 모델을 활용하면 문서를 구성하고 있는 단어 벡터의 표현력이 좋아지며, 새로운 단어의 임베딩 생성 또한 가능하므로 새로운 단어가 많이 등장하는 문서의 경우 더욱 유리하다. 임베딩 생성기의 결과인 단어 벡터를 Recurrenct Neural Network에 입력하여 문서 분류기를 학습 시킨다면, 학습 말뭉치에 편향성이 적은 범용적인 문서 분류기 학습이 가능할 것이다.

기존의 단어 빈도수 기반 문서 분류는 단순히 문서 내에 나온 단어의 빈도를 사용하여 문서 분류를 하였는데, 기존 방법보다 문서가 갖고 있는 특징을 더 정교하게 표현 가능하므로, 문서 분류 성능을 높일 것이라 기대한다.

기존 연구

word-embedding

Skip-gram은 비슷한 위치에 등장하는 단어의 의미는 유사함을 이용하여 학습을 한다. 중심 단어가 주어졌을 때, 주변 단어가 나올 확률은 최대화 되고, 주변 단어가 아닌 단어들이 나올 확률은 최소화 되는 방향으로 학습을 한다. 아래와 같은 목적 함수가 최소화 되도록 학습한다.

$$\mathcal{J} = \log P(w_{c-m}, \cdots, w_{c-1}, w_{c+1}, \cdots, w_{c+m} | w_c) \quad (1)$$

$$\mathcal{J} = -\sum_{j=0, j\neq m}^{2m} \log P(u_{c-m+j} | v_c) \quad (2)$$

$w_i \in \mathcal{R}^{|V|}$, $\mathcal{V} \in \mathcal{R}^{n\times|V|}$, $\mathcal{U} \in \mathcal{R}^{n\times|V|}$, v_i :i-th column of $\mathcal{V}$, μ_i :i-th column of $\mathcal{U}$ 이다.

Skip-gram 모델의 단점은 단어 벡터 학습이 윈도우 내에서만 이뤄지기 때문에 말뭉치 전체에서 동시 등장한 정보는 반영되기 어렵다는 것이다. 이 문제를 해결하고자, Glove에서는 목적함수를 임베딩 된 두 단어 벡터의 내적이 말뭉치 전체에서의 동시 등장 확률 로그 값이 되도록 학습한다. 따라서, 유사도 측정 뿐만 아니라 말뭉치 전체의 통계 정보도 반영할 수 있게 된다. Glove의 목적함수는 아래와 같다.

$$\mathcal{J} = -\sum_{i=1}^{W}\sum_{j=1}^{W} f(X_{ij})(v_j^T u_i - \log(X_{ij}) \quad (3)$$

Skip-gram 및 Glove는 현재 많이 쓰이고 있는 임베딩 방법으로, 단어의 의미론적 특징을 표현하기는 쉽지만, 형태 정보를 반영하는 데에는 한계가 있다. 또한 이전에 학습시키지 않았던 새로운 단어를 표현하지 못하는 문제점을 갖고 있다.

FastText는 이를 해결하고자, 단어 벡터를 부분 단어(n-grams)의 백터들을 합하여 표현한다. 따라서, 드문 단어들을 좀 더 잘 표현할 수 있으며, 노이즈가 많은 말뭉치에 강점을 지니게 된다.

Skip gram, Glove 및 FastText 모델을 학습시킬 때는 negative sampling 방법을 사용한다. Negative sampling을 사용하지 않고 위와 같은 모델을 학습시키면 단어의 수에 따라서 계산 량이 매우 커지게 되며, 더불어 학습도 잘 되지 않는다. 따라서 window에 포함되지 않은 단어에 대해서는, 말뭉치의 단어 빈도수를 고려하여 sampling을 하고, 이러한 sample들만 loss function 계산에 이용하여 효율적으로 loss를 계산할 수 있다. Negative sampling을 사용한 목적 함수는 아래와 같이 정의된다.

$$\mathcal{J} = -\log\sigma\left(u_{c-m+j}^T v_c\right) - \sum_{k=1}^{K} log\sigma(-\tilde{u}_k^T v_c) \quad (3)$$

문서 분류

문서 분류를 위해서는 문서를 벡터로 표현해야한다. 문서를 벡터로 표현하는 방법은 다양하지만, 가장 많이 쓰이는 방식은 count 기반의 bag-of-words 방법이다. 말뭉치의 단어 개수가 문서 벡터의 차원이 되며, 벡터의 값은 각 단어가 등장한 빈도이다. Bag-of-words로 표현한 벡터를 이용하여 SVM, Naïve Bayes 등의 분류기를 학습시켜 문서 분류기로 이용한다.

최근에는 Neural Network를 이용한 문서 분류 방법도 연구되고 있다. Y Zhang et al.(2015)와 Y Kim (2014)와 이다. 두 논문 모두 pre-trained word vector를 CNN 층을 통과시켜 문서를 분류했다. 임베딩 된 단어와 Neural Network를 이용하여 문서를 분류하는 것이 단순히 문서의 count vector를 이용하는 것보다 더 좋은 성능을 보였다.

단어 임베딩 생성기(RNN Based Skip-gram)

본 논문에서는 Recurrent Neural Network(RNN) 모델

과, skip-gram모델을 결합한 모델을 제안한다.

RNN에 중심 단어 및 주변 단어를 글자 단위로 입력을 하여 얻은 결과인 h_T를 각 단어의 형태적 특징 임베딩으로 활용한다. 또한, 중심 단어와 주변 단어가 함께 나올 확률은 높게, 중심 단어와 주변 단어가 아닌 단어가 함께 나올 확률은 낮게 학습하여 의미론적 특징도 포함하도록 단어 임베딩을 학습한다.

글자 단위로 입력하기 때문에 단어의 형태학적 정보를 학습할 수 있으므로 단어보다 더 작은 단위인 접두사, 접미사 및 시제 등의 성질 표현이 가능할 것이라 기대한다. 또한, 글자 단위로 텍스트를 입력하기 때문에 stemming, lemmatization 등의 전처리 필요성이 줄어들게 된다.

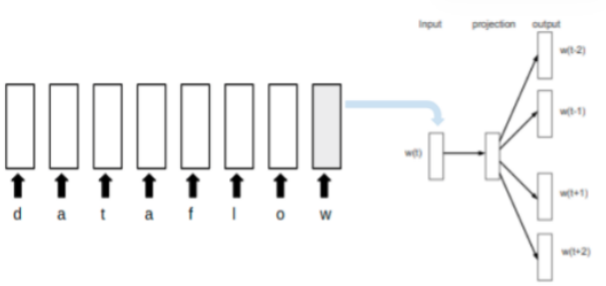

그림 1 – 제안하는 모델의 전체 모식도

RNN 임베딩

빈번히 쓰이는 RNN 모델로는 GRU(Gated Recurrent Unit), LSTM(Long Short Term Memory) 가 있으며, 본 논문에서는 LSTM을 사용했다.

위 그림 1 과 같이 dataflow라는 단어를 LSTM에 입력시키는데, d, a, t, a, f, l, o, w를 embedding하여 입력한다. 위의 예시와 같은 경우, sequence length는 8이며 마지막 hidden state인 h_T를 단어 임베딩의 결과로 얻는다. 마지막 hidden vector에는 단어의 형태학적 특징이 encoding되었음을 기대한다.

$$h_t = \sigma(W_{ce}^{hh} h_{t-1} + W_{ce}^{hw} x_t) \quad (4)$$

$x_t \in \mathcal{R}^d$, $W_{ce}^{hx} \in \mathcal{R}^{D_h \times d}$, $W_{ce}^{hh} \in \mathcal{R}^{D_h \times D_h}$, $h_{t-1} \in \mathcal{R}^{D_h}$ 이다. W_{ce}^{hx}와 W_{ce}^{hh}가 학습 가능한 parameter이다.

중심 단어, 주변 단어 및 negative sampling 각 단어의 분산 표현을 LSTM을 통해 얻는다. 중심 단어 LSTM과 주변 단어 및 negative sampling LSTM, 총 2개의 LSTM을 각각 학습한다. 중심 단어 LSTM은 W_{ce}로 표기하였고, 위 수식이 주변 단어 LSTM에도 똑같이 적용된다.

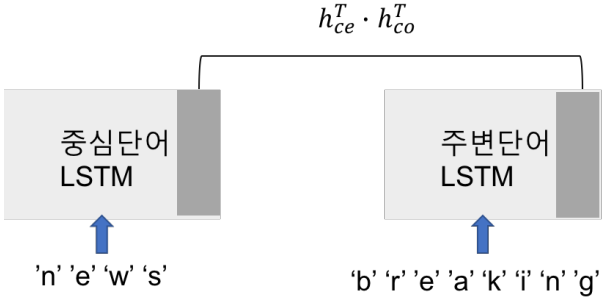

그림 2 - 중심단어 LSTM 및 주변 단어 LSTM

예를 들어, 'breaking news analysis is important for national security' 문장에서 news가 중심 단어로 설정하고, window size를 1로 설정하면, (news, breaking), (news, analysis)와 같은 (중심단어, 주변단어) pair를 얻는다. 그림 2와 같이 각각을 LSTM에 입력하여 계산한다.

LSTM의 parameter가 전체 단어 임베딩 정보를 학습할 수 있도록 D_h를 일반적인 LSTM의 hidden vector 보다는 크게 설정한다.

최종적으로, LSTM의 output을 각각 fully-connected layer를 통과시켜 단어의 임베딩을 얻는다.

목적함수

본 논문에서 제안한 모델을 학습시킬 때의 목적 함수는 기존 연구에서 언급한 Skip-gram의 것과 유사하다. Skip-gram에서는 단어를 보고 주변 단어를 예측하게 된다. 중심 단어는 w_c, 주변 단어는 $w_{c-m}, \cdots, w_{c-1}, w_{c+1}, \cdots w_{c+m}$으로 표기한다.

기존의 Skip-gram에서는 이미 학습이 된 단어 임베딩을 look-up하여 w를 나타낸 반면에, 본 논문에서 제안하는 모델에서는 RNN의 결과물인 h_T로 각 단어들을 나타낸다.

최종 목적 함수는 (1) 및 (2)와 유사하지만, w가 h_T로 바뀌었다.

$$\mathcal{J} = -\log P(h_T^{c-m}, \cdots, h_T^{c-1}, h_T^{c+1}, \cdots, h_T^{c+m} | h_T^c) \quad (5)$$

h_c^T는 중심 단어의 LSTM 마지막 hidden vector h_T이다. 단어 임베딩 생성기 또한 skip-gram에서 제안한 negative sampling 방법을 이용하여 학습한다.

문서 분류기

본 논문에서 제안한 단어 임베딩 생성기를 pre-training시켜, 문서 내 단어 분산 표현을 얻고 이를 문서 분류 RNN에 통과시켜 문서 분류기를 학습할 수 있다. 학습 말뭉치에 없는 단어의 임베딩을 만들 수 있으며, RNN 분류기를 사용하여, 문서의 sequential한 특징도 학습하여 더 정확한 문서 분류가 가능할 것을 기대한다.

실험

단어 임베딩 생성기 학습

실험 세부사항

실험에 이용한 dataset은 English Wiki Dump Dataset이다. 실험의 Baseline으로 사용한 skip-gram의 경우 전체 dataset(32GB)의 약 10%만 사용하여 학습을 진행했다. 반면에, 본 모델에서 제안하는 RBSG(RNN Based Skip-gram)은 전체 dataset의 약 3%만 사용하여 학습을 진행했다. 두 모델 모두 학습한 epoch은 1회이다.

임베딩 생성기의 평가를 위해, 모델의 결과와 사람의 판단 사이의 상관관계를 계산하여 단어 임베딩 성능을 확인하는 similarity dataset을 사용했다. MTurk(2012), MEN(2012), WS353(2001), RW(rare word dataset, 2013), SIMLEX99(2014) 총 5개의 similarity dataset을 활용했다.

Dataset 전 처리의 경우 다음과 같은 parameter를 설정했다. 7번 negative sampling을 했고, 등장 빈도수가 높은 단어들을 제거하기 위해 10^{-4}의 threshold 를 사용하여 subsampling을 했다. 또한 5번 이하로 드물게 등장하는 단어는 제거 했다.

모델 학습의 parameter는 다음과 같다. Baseline 실험으로 사용한 skip-gram의 경우, embedding size는 300, learning rate는 0.025로 설정한 다음 10 step마다 이전 learning rate의 0.9배가 되도록 scheduling 했다.

본 논문에서 제안한 RBSG(RNN Based Skip-gram)의 경우, character embedding size 128, LSTM hidden size 600, learning rate는 0.002고 skip-gram과 같은 scheduling을 적용했다.

모델 평가 시, Inference에 사용한 RBSG의 LSTM 모듈은 중심 단어 LSTM과 주변 단어 LSTM 두개 중 중심 단어 LSTM을 사용했다.

두 모델 다 optimizer로는 Adam을 이용했다.

결과

표 1 human judgement와 모델 별 similarity score의 상관관계

Model/ dataset	MTurk	MEN	WS353	RW	SIMLEX999
Skip-gram	0.021	0.129	0.162	-0.045	-0.012
RBSG	0.012	0.003	0.004	**0.072**	**0.013**

표 1에서 볼 수 있듯이, 전반적으로 skip-gram 보다 상관관계가 높지 않았지만, SIMPLES999와 Rare Words dataset에 대해서는 논문에서 제안한 RBSG 모델이 더 좋은 점수를 냈다. 특히, Rare Words에서 더 좋은 점수를 보였는데, 이 결과를 통해 임베딩 생성기가 OOV 문제를 완화시킨 다는 것을 알 수 있다.

결과

결론

본 논문에서 제안한 모델이 드문 단어들이 많은 Rare words dataset에서 skip-gram 보다 더 높은 점수를 얻었다. 기존 단어 분산 표현에서 가장 큰 문제였던 OOV 문제를 해결할 수 있다는 가능성을 보였다.

본 논문을 발전시키기 위해서 몇 가지 부분에 대한 후속 연구가 필요하다. 첫 번째로, 영어 외 다양한 언어에 대해 단어 임베딩 생성기를 학습시킬 필요가 있다. 영어는 아랍어, 독일어, 러시아어에 비해 형태학적 특징의 중요도가 낮은 언어 이므로, 단어의 형태가 중요한 역할을 하는 언어에 본 논문에서 제안한 모델이 성능 향상을 보일 것이라 예상한다. 한국어 또한, 단어의 형태학적 특징이 중요한 언어로 단어를 이루고 있는 자음과 모음으로 분리하여 임베딩 할 수 있다.

또한 자원 부족의 문제로 학습시킨 dataset의 양이 다른 기존 연구에 비해 현저히 적었다. 여러 대의 컴퓨터를 이용하여 모델을 분산 학습시켜 더 많은 데이터를 학습시키는 방향으로 논문을 발전시킬 수 있다.

References

[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] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

[3] Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1532-1543).

[4] Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2016). Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

[5] Zhang, Y., & Wallace, B. (2015). A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. *arXiv preprint arXiv:1510.03820*.

[6] Kim, Y. (2014). Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.