Fasttext Review

Summary

단어를 벡터로 바꾸는 임베딩 작업은 nlp 과제에서 유용게 쓰인다. 하지만 유명한 모델은 단어의 형태론(morphology)을 무시하여 각 단어에 distict vector을 할당한다. 이는 큰 단어 사전과 희귀한 단어를 가지는 언어에 특히 한계를 가진다.

예를 들어 불어나 스페인어는 대부분의 동사가 40개 이상의 변형된 형태를 가진다. 즉, 불어는 동사를 행하는 주체가 남성, 여성, 사물인지의 여부에 따라 형태가 모두 다르며 시제 또한 매우 여러가지가 있다. 이 중에는 많이 쓰이는 형태도 있겠지만, 드물게 쓰이는 형태가 더 많을 것이다. 같은 동사에서 파생됐지만 그 외형이 다르다고 해서 상이한 벡터로 표현하는 것은 적절하지 않다. 하지만 이렇게 한 동사에서 파생된 단어들은 기본적으로 공유하는 어간이 있을 것이다. Fasttext은 이 점을 이용한다.

Fasttext은 skip gram 모델에 기반하면서각 단어는 a bag of character n-grmas으로 표현된다. 한 벡터 표현은 각각의 character n-gram과 연관되어 있고 단어들은 이러한 벡터 표현의 합(sum)으로 표현된다. Fasttext는 큰 코퍼스로부터 빠르게 학습하면서 언어의 morphology 관계를 보존한다. 학습이 빠르다는 점은 다른 모델들과 구분되는 확연한 장점인데, 텍스트 데이터는 그 크기가 매우 크고 이를 학습하는데 오래 걸리는 모델들이 다수 있기 때문이다.

General Model

Fasttext은 skip gram 모델에 기반하여 만들어졌으므로 우선 이에 대하여 간략하게 살펴보자.

단어 사전의 크기가 W로 주어질 때, 각 단어는 $w \in \{1, \cdots, W\}$ 인덱스에 의해서 표현되고 목표는 각 단어 w에 대한 벡터 표현을 찾는 것이다. 이때, 이러한 벡터 표현은 문ㅁ냑에서 나타날 단어를 잘 예측하도록 훈련된다. 즉, 거대한 훈련 코퍼스가 $w_1, \cdots, w_T$의 단어 sequence로 주어지면 skip gram 모델의 목적은 아래 log likelihood를 최대화하는 것이다.

$$\sum_{t=1}^{T} \sum_{c \in \mathcal{C}_t} log\, p(w_c \mid w_t)$$

여기서 $\mathcal{C}_t$는 단어 w_t 주위를 둘러싸는 다른 단어들의 인덱스이다. w_t가 주어진 상태에서 주변 단어들인 w_c을 관찰할 확률은 앞서 언급한 단어 벡터들로 parameterize 된다. 이제 (word, context)을 실수($\mathbb{R}$) 인 점수로 mapping 해주는 함수 s를 생각하자. 확률로 가능한 선택지 중 하나는 softmax이다.

$$p(w_c \mid w_t) = \frac{e^{s(w_t, w_c)}}{\sum_{j=1}^{W} e^{s(w_t, w_j)}}$$

하지만 이러한 모델은 w_t가 주어질 때, 하나의 문맥 단어인 w_c가 나올 확률을 예측한다.

문맥 단어를 예측하는 문제는 독립적인 이진(binary) 분류 문제로 바꿔 생각할 수 있다. 따라서 목표는 독립적으로 문맥 단어의 존재 또는 부재를 예측하는 것이다. w_t에 대해 문맥 단어들을 positive examples로 그렇지 않은 단어들을 negative examples로 간주한다. 예를 들어 아래와 같은 문장을

살펴보자.

자연어처리는 어렵지만 흥미롭고 발전 가능성을 갖추었다.

이 문장에서 w_t를 발전이라고 가정하고 윈도우는 2로 설정하자. 이때 positive, negative 샘플은 아래와 같다.

positive 샘플		negative 샘플	
t	c	t	c
발전	어렵지만	발전	어렵지만
발전	흥미롭고	발전	흥미롭고
발전	가능성을	발전	가능성을
발전	갖추었다.	발전	갖추었다.

만약 모든 negative 샘플을 사용한다면 계산량이 매우 많아질 것이므로 이 중 n개만 샘플하여 사용한다. 따라서 w_t에 대한 하나의 문맥 단어인 w_c에 대해서 n개의 negative 샘플을 고려한 binary logistic loss는 아래와 같이 w_c가 나타날 확률과 n개의 negative 샘플이 나타날 확률을 로지스틱 함수를 통해서 곱하고 이에 -을 취한 negative log likelihood이다.

$$
\begin{aligned}
binary\,logistic\,loss &= -log\ (positive \times negatives) \\
&= -log\left[\left(1+e^{-s(w_t,w_c)}\right)^{-1}\prod_{n\in\mathcal{N}_{t,c}}\left\{1-\left(1+e^{-s(w_t,w_c)}\right)^{-1}\right\}\right] \\
&= log\left(1+e^{-s(w_t,w_c)}\right)-\sum_{n\in\mathcal{N}_{t,c}} log\left\{1-\frac{1}{1+e^{-s(w_t,w_c)}}\right\} \\
&= log\left(1+e^{-s(w_t,w_c)}\right)-\sum_{n\in\mathcal{N}_{t,c}} log\left\{\frac{e^{-s(w_t,w_c)}}{1+e^{-s(w_t,w_c)}}\right\} \\
&= log\left(1+e^{-s(w_t,w_c)}\right)+\sum_{n\in\mathcal{N}_{t,c}} log\left(\frac{1+e^{-s(w_t,w_c)}}{e^{-s(w_t,w_c)}}\right) \\
&= log\left(1+e^{-s(w_t,w_c)}\right)+\sum_{n\in\mathcal{N}_{t,c}} log\left(1+e^{s(w_t,w_c)}\right)
\end{aligned}
$$

여기서 $\mathcal{N}_{t,c}$는 negative 샘플의 집합이다. logistic loss function인 $\ell : x \longmapsto log(1+e^{-x})$을 이용, 전체 w_t에 대해서 위 negative log likelihood을 다시 쓰면 아래와 같다.

$$
\sum_{t=1}^{T}\left[\sum_{c\in\mathcal{C}_t}\ell\left(s\left(w_t,w_c\right)\right)+\sum_{n\in\mathcal{N}_{t,c}}\ell\left(-s\left(w_t,n\right)\right)\right]
$$

이제 scoring function인 s에 대해서 생각해보자. $s(w_t,w_c)$는 input으로 단어, 문맥 단어가 들어가고 output으로 어떤 score을 반환한다. scoring function의 가능한 한 예는 words vectors을 사용하는

것이다. 단어 w_t, w_c의 벡터를 $\mathbf{u}_{w_t}, \mathbf{v}_{w_c}$라고 하자. w_t, w_c간의 score는 이 두 벡터의 내적으로 계산할 수 있다. 즉,

$$s(w_t, w_c) = \mathbf{u}_{w_t}^T \mathbf{v}_{w_c}$$

여태까지 설명한 모델은 word2vec의 negative sampling이다.

Subword Model

각 단어에 대해서 별개의 벡터 표현을 함으로써, skip gram 모델은 단어의 내적 구조를 무시한다. 여기서는 이러한 정보를 고려하기 위해서 다른 scoring function s을 제안한다.

각 단어 w는 a bag of character n-gram으로 표현된다. 여기에 특별한 boundary 표현법인 <와 > 을 단어의 처음과 끝에 추가함으로써 다른 문자 sequences로부터 접두사와 접미사를 구별한다. 또한 단어 그 자체인 w도 n-gram의 일부로 포함시킨다. 예를 들어서 $where$에 대해 3-grams의 예시를 살펴보자. 위에서 정의한 규칙을 이용하여

$$< wh,\ whe,\ her,\ ere,\ re >, < where >$$

총 6가지의 3-grams로 표현할 수 있다. 여기서 단어 her로부터 나온 $< her >$은 $where$의 3-gram인 her과 다름에 주목하자.

이 접근법은 매우 간단하고 다른 종류의 n-gram도 고려될 수 있다. 예를 들어 모든 접두사와 접미사를 취하는 것이 한 예이다.

크기가 G인 n-grams 사전이 주어졌다고 하자. 단어 w가 주어질 때, 이를 $\mathcal{G}_w \subset \{1, \cdots, G\}$라고 하자(단어 w에 나타나는 n-gram의 집합) $g \in \mathcal{G}_w$인 g에 대해서 g를 표현하는 벡터를 $\mathbf{z}_g$라고 하자. Fasttext는 이를 이용하여 scoring function을 아래와 같이 정의한다.

$$s(w_t, w_c) = \sum_{g \in \mathcal{G}_w} \mathbf{z}_g^T \mathbf{v}_c$$

Fasttext의 이러한 scoring은 간단하고 단어들에 걸쳐서 표현법을 공유한다. 따라서 희귀한 단어들에 대해 꽤 괜찮은 언어를 배우게 된다.

Reference

1. Bag of Tricks for Efficient Text Classification, Armand Joulin et al., 2016
2. EnrichingWord Vectors with Subword Information, Piotr Bojanowski et al., 2017
3. 한국어 임베딩, 이기창