Efficient Vector Representation for Documents through Corruption

Minmin Chen (2017)

Presenter: Sarah Kim

2018.04.26

Introduction

- Document Vector through Corruption (Doc2VecC):

  each document as a simple average of word embeddings of all words in the document.

- Motivation (Mikolov et al., 2013):

  (Ex. 1) vec("Rusia")+vec("river") $\approx$ vec("Volga River")

  (Ex. 2) vec("king")$-$vec("man")+vec("women") $\approx$ vec("queen").

- During learning, randomly remove words from a document.

Notations

- Notations:
  - $\mathcal{D} = \{D_1, \ldots, D_n\}$: a training corpus of size n, and each D_i consists of words $w_i^1, \ldots, w_i^{T_i}$;
  - V: the vocabulary used in the training corpus, of sizes v;
  - $\mathbf{x} \in \mathbb{R}^{v \times 1}$: BoW of a document, where $x_j = 1$ iff word j does appear in the document;
  - $\mathbf{c}^t \in \mathbb{R}^{v \times 1}$: BoW of the local context $w^{t-k}, \ldots, w^{t-1}, w^{t+1}, \ldots, w^{t+k}$ at the target position t, $c_j^t = 1$ iff word j appears within the sliding window of the target;
  - $\mathbf{U} \in \mathbb{R}^{h \times v}$: the projection matrix from the input space to a hidden space of size h;
  - $\mathbf{V}^\top \in \mathbb{R}^{v \times h}$: the projection matrix from the hidden space to ouput.

Method

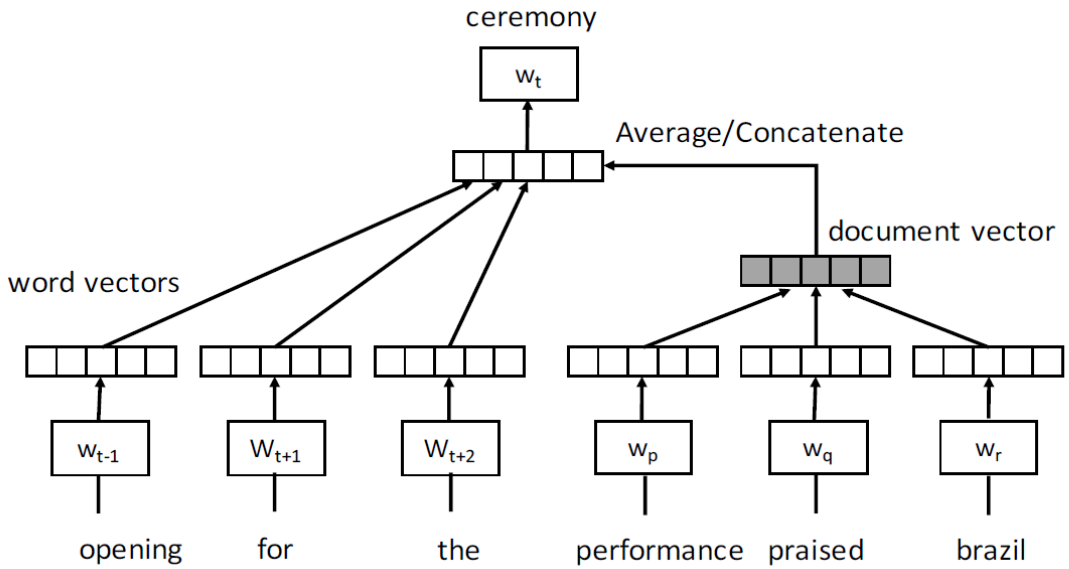


Figure 1 : A new framework for learning document vectors.

- Embeddings of neighboring words → local context
- Vector representation of the entire document → global context.

Method

- To generate a global context at each update, we use a unbiased mask-out/drop-out corruption:

$$\tilde{x}_d = \begin{cases} 0, & \text{with probability } q, \\ \frac{x_d}{1-q}, & \text{otherwise.} \end{cases}$$

- Doc2VecC:

$$P(w^t|\mathbf{c}^t, \tilde{\mathbf{x}}) = \frac{\exp(\mathbf{v}_{w^t}^\top(\mathbf{U}\mathbf{c}^t + \frac{1}{T}\mathbf{U}\tilde{\mathbf{x}}))}{\sum_{w' \in V} \exp(\mathbf{v}_{w'}^\top(\mathbf{U}\mathbf{c}^t + \frac{1}{T}\mathbf{U}\tilde{\mathbf{x}}))},$$

where T is the length of the document.

Method

- $\mathbf{U}$ and $\mathbf{V}$ are learned to minimize the loss:

$$\ell = -\sum_{i=1}^{n}\sum_{t=1}^{T_i} f(w_i^t, \mathbf{c}_i^t, \tilde{\mathbf{x}}_i^t) = -\sum_{i=1}^{n}\sum_{t=1}^{T_i} \log P(w_i^t | \mathbf{c}_i^t, \tilde{\mathbf{x}}_i^t)$$

- Given the learned projection matrix $\mathbf{U}$, we represent each document as

$$\mathbf{d} = \frac{1}{T}\sum_{w \in D} \mathbf{u}_w.$$

Corruption as data-dependent regularization

- Using Taylor expansion of $f(w, \mathbf{c}, \tilde{\mathbf{x}})$ w.r.t. $\tilde{\mathbf{x}}$ up to the second-order, we can see that Doc2VecC intrinsically minimizes

$$\ell \approx -\sum_{i=1}^{n}\sum_{t=1}^{T_i} f(w_i^t, \mathbf{c}_i^t, \mathbf{x}_i) + \frac{q}{1-q}\sum_{j=1}^{v} R(\mathbf{u}_j),$$

where the second term is a data-dependent regularization,

$$R(\mathbf{u}_j) \propto \sum_{i=1}^{n}\sum_{t=1}^{T_i} x_{ij}^2 \Big[\sigma_{w_i^t,\mathbf{c}_i^t,\mathbf{x}_i}(1-\sigma_{w_i^t,\mathbf{c}_i^t,\mathbf{x}_i})(\frac{1}{T}\mathbf{v}_{w_i^t}^\top \mathbf{u}_j)^2 + \sum_{w'\sim P_v} \sigma_{w',\mathbf{c}_i^t,\mathbf{x}_i}(1-\sigma_{w',\mathbf{c}_i^t,\mathbf{x}_i})(\frac{1}{T}\mathbf{v}_{w'}^\top \mathbf{u}_j)^2\Big],$$

where P_v stands for a uniform distribution over the terms in the vocab, and $\sigma_{w,\mathbf{c},\mathbf{x}} = \sigma(\mathbf{v}_w^\top(\mathbf{U}\mathbf{c} + \frac{1}{T}\mathbf{U}\mathbf{x}))$.

Experiments

Sentiment analysis

- For sentiment analysis, we use the IMDB movie review dataset. It contains 100,000 movies reviews categorized as either positive or negative.

Model	Error rate % (include test)	Error rate % (exclude test)
Bag-of-Words (BOW)	12.53	12.59
RNN-LM	13.59	13.59
Denoising Autoencoders (DEA)	11.58	12.54
Word2Vec + AVG	12.11	12.69
Word2Vec + IDF	11.28	11.92
Paragraph Vectors	10.81	12.10
Skip-thought Vectors	-	17.42
Doc2VecC	**10.48**	**11.70**

Figure 2 : Classification error of a linear classifier trained on various document representations.

Experiments

Sentiment analysis

Table 2: Learning time and representation generation time required by different representation learning algorithms.

Model	Learning time	Generation time
Denoising Autoencoders	3m 23s	7s
Word2Vec + IDF	2m 33s	7s
Paragraph Vectors	4m 54s	4m 17s
Skip-thought	2h	2h
Doc2VecC	4m 30s	7s

Table 3: Words with embeddings closest to 0 learned by different algorithms.

Word2Vec	harp(118) distasteful(115) switzerland(101) shabby(103) fireworks(101) heavens(100) thornton(108) endeavor(100) dense(108) circumstance(119) debacle(103)
ParaVectors	harp(118) dense(108) reels(115) fireworks(101) its'(103) unnoticed(112) pony(102) fulfilled(107) heavens(100) bliss(110) canned(114) shabby(103) debacle(103)
Doc2VecC	,(1099319) .(1306691) the(1340408) of(581667) and(651119) up(49871) to(537570) that(275240) time(48205) endeavor(100) here(21118) way(31302) own(13456)

Document Embeddings via Recurrent Language Models

Giel, Andrew, and Ryan Diaz (2016)

DRNLLM

- Recurrent Neural Network Language Model (RNNLM) trained by a document vector into the calculation of the hidden state and prediction at each time step.
- Mappings
  - L: Word embedding matrix, with each word has a unique column within L.
  - D: Document matrix, with each document mapped to a unique column within D.

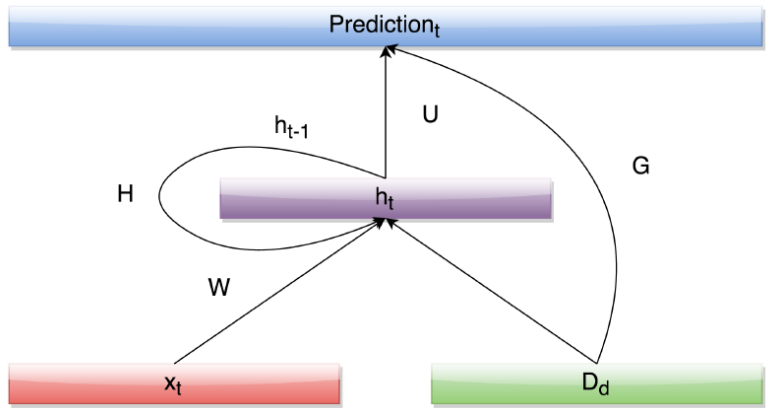
DRNLLM

- DRNNLM is given a series of words $x_{m-n}, \ldots, x_m$ with the goal of predicting x_{m+1}.
- The values of hidden layers and output with document i are defined as follows

$$h_t = \sigma(WL_{x_t} + Hh_{t-1} + D_i + b_h)$$

$$y_t = g(Uh_t + GD_i + b),$$

where $\sigma(z)$ is the sigmoid function and $g(p)$ is the softmax function.



- Minimize the cross-entropy loss for L, D, W, H, U, G, b_h, b.
- For a new corpus of documents, we expand D. For each new document train the corresponding column vector by minimizing cross-entropy, and only update D.

Experiments

- We used the 20 Newsgroups dataset which consists of 20000 short documents grouped into 20 distinct categories.
- The document vectors produced from running DRNLLM do not show clear separability based on the class of the document.

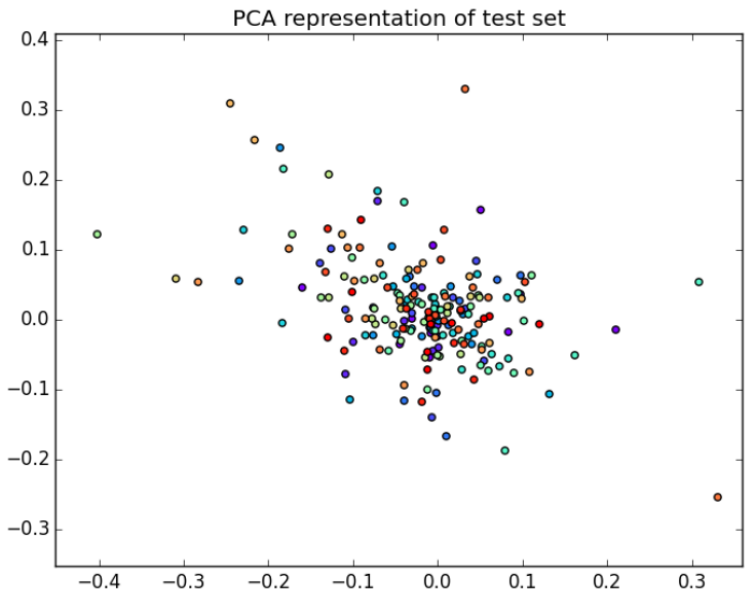


Figure 3 : Sampling of 200 document vectors plotted via PCA