Векторные представления слов: word2vec

Скачков Николай Андреевич

ВМК МГУ

2017/14/02

Мотивация

Как мы умеем представлять слова из словаря Ω в виде объектов?

Мотивация

Как мы умеем представлять слова из словаря Ω в виде объектов?
One-hot-encoding:

$$w \longrightarrow v_w : \quad v_w^i = [\#w = i]$$

У этого способа кодирования очень много недостатков!

Мотивация

Как мы умеем представлять слова из словаря Ω в виде объектов?
One-hot-encoding:

$$w \longrightarrow v_w : \quad v_w^i = [\#w = i]$$

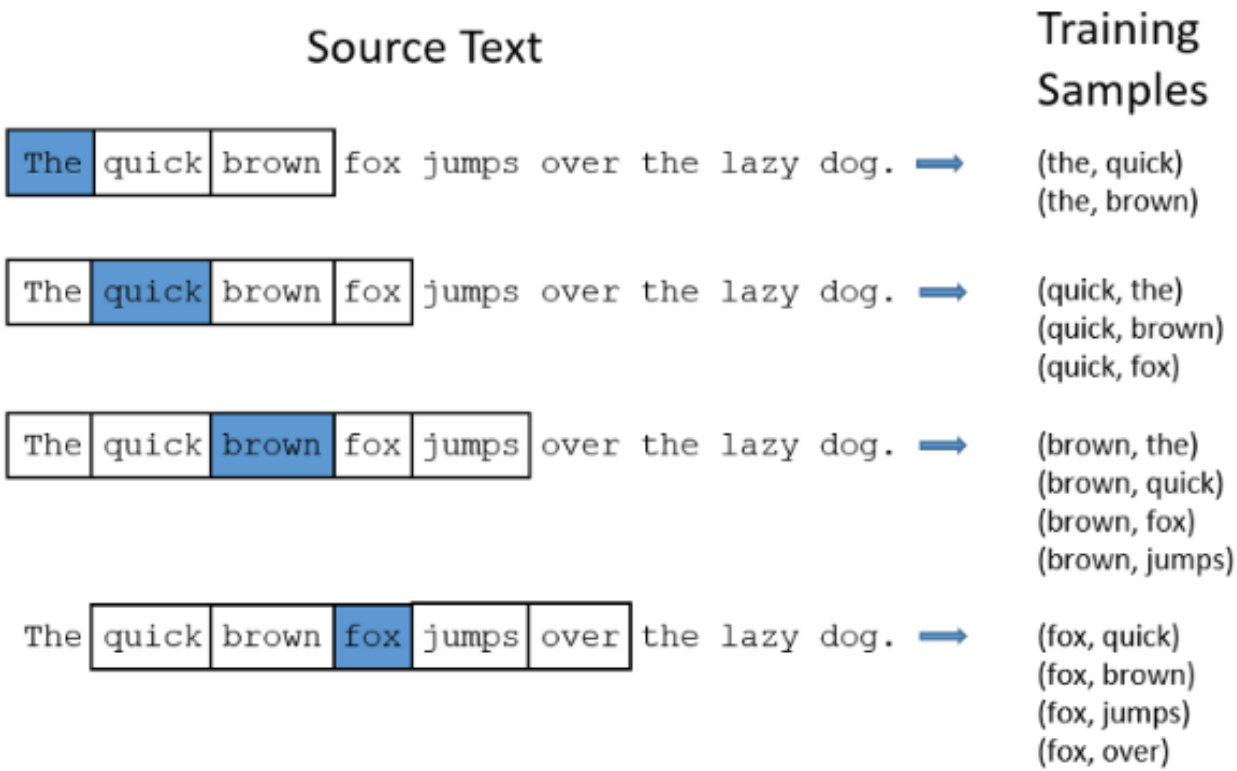
У этого способа кодирования очень много недостатков!
Хотим получить:

- представление слов в пространстве H невысокой размерности
- в данном пространстве H близкие по смыслу слова находятся рядом
- пространство H – линейное:

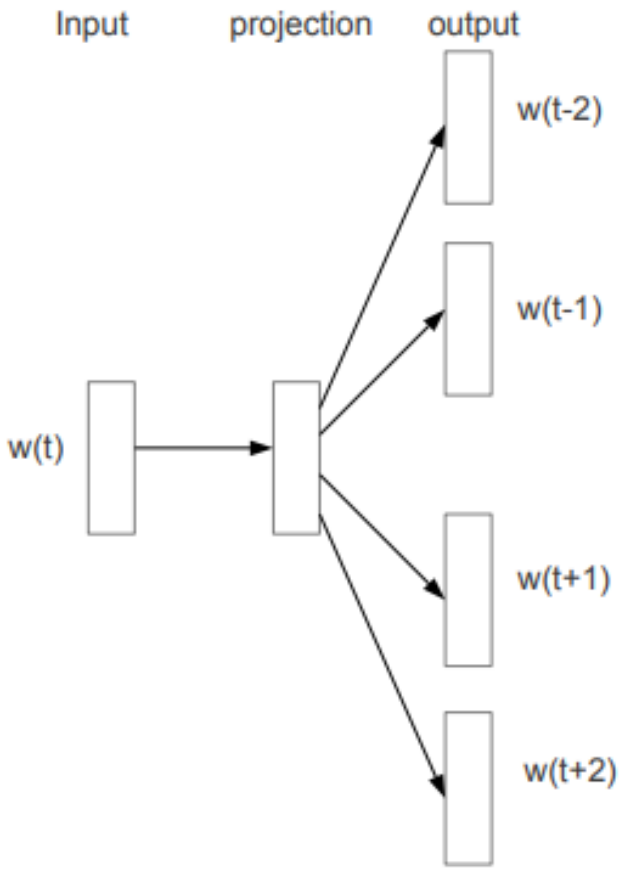
  vec(Madrid) - vec(Spain) + vec(France) = vec(Paris)

Использование контекста

- **Идея:** Близкие слова имеют схожие контексты.
- Построим модель, которая будет хорошо предсказывать слова по контексту.



Skip-gram



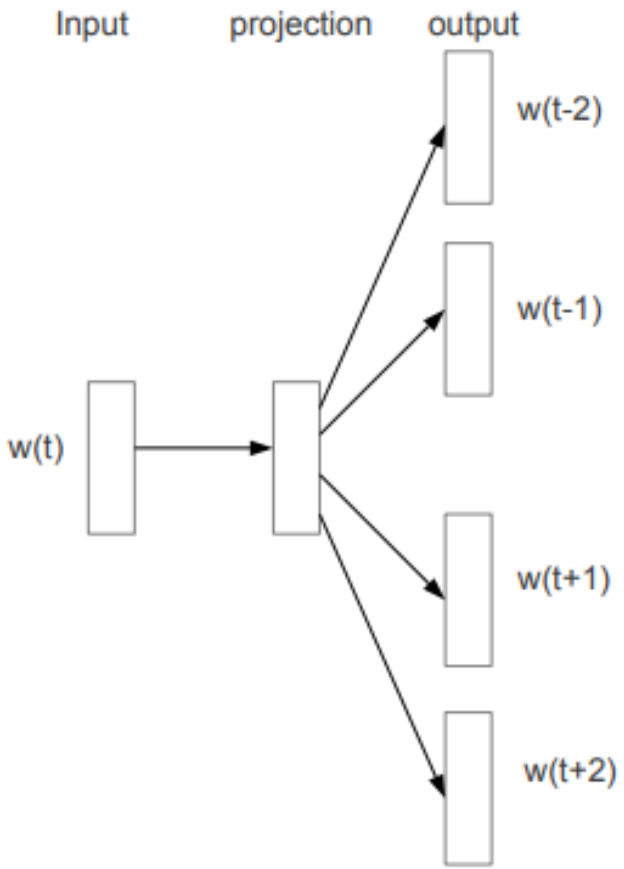
- Параметры W, W':

  $v_w = W_w$ – input vector.

  $v'_w = W'^{T}_w$ – output vector.

T. Mikolov Distributed Representations of Words and Phrases and their Compositionality, 2013

Skip-gram



- Параметры W, W':

  $v_w = W_w$ – input vector.

  $v'_w = {W'_w}^T$ – output vector.

- Подсчет условной вероятности (softmax):

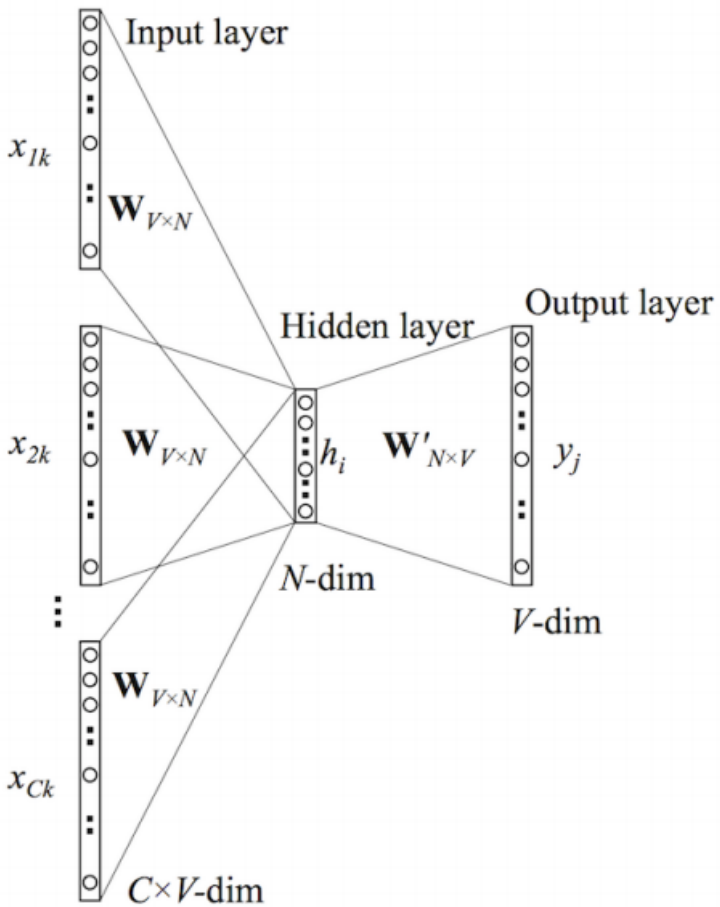
$$p(w_O|w_I) = \frac{\exp({v'_{w_O}}^T v_{w_I})}{\sum_{w \in \Omega} \exp({v'_w}^T v_{w_I})}$$

- Функция потерь:

$$L(w_t) = -\sum_{|j|=1}^{k} \log p(w_{t+j}|w_t)$$

T. Mikolov Distributed Representations of Words and Phrases and their Compositionality, 2013

Continuous bag-of-words



- Параметры W, W':

  $v_w = W_w$ – input vector.

  $v'_w = {W'_w}^T$ – output vector.

- Подсчет условной вероятности (softmax):

$$h = \frac{1}{2k-2} \sum_{|j|=1}^{k} v_{w_j}$$

$$p(w_O|h) = \frac{\exp({v'_{w_O}}^T h)}{\sum_{w\in\Omega} \exp({v'_w}^T h)}$$

- Функция потерь:

$$L(w_t) = -\sum_{|j|=1}^{k} \log p(w_t|h)$$

T. Mikolov Efficient Estimation of Word Representations in Vector Space, 2013

Skip-gram vs CBOW

Skip-gram:

+ Хорошо предсказывает контексты по редким словам, так как модель предсказывает по ним их соседей.
— Обучается в разы дольше, чем CBOW.

CBOW:

+ Быстрое обучение
— Редкие слова почти не влияют на обучение модели и выучиваются значительно хуже.

Model	Vector Dimensionality	Training words	Accuracy [%]			Training time [days]
			Semantic	Syntactic	Total	
3 epoch CBOW	300	783M	15.5	53.1	36.1	1
3 epoch Skip-gram	300	783M	50.0	55.9	53.3	3

Оптимизация обучения

- Количество параметров:

$$|\Omega| = 10000, \quad |h| = 300 \quad \longrightarrow \quad |W| = 3 \cdot 10^9$$

- Вычисление градиента softmax происходит за $O(|\Omega|)$

Оптимизация обучения

- Количество параметров:

$$|\Omega| = 10000, \quad |h| = 300 \quad \longrightarrow \quad |W| = 3 \cdot 10^9$$

- Вычисление градиента softmax происходит за $O(|\Omega|)$

Необходимо оптимизировать вычисление функции потерь!
Для этого изменим ее:

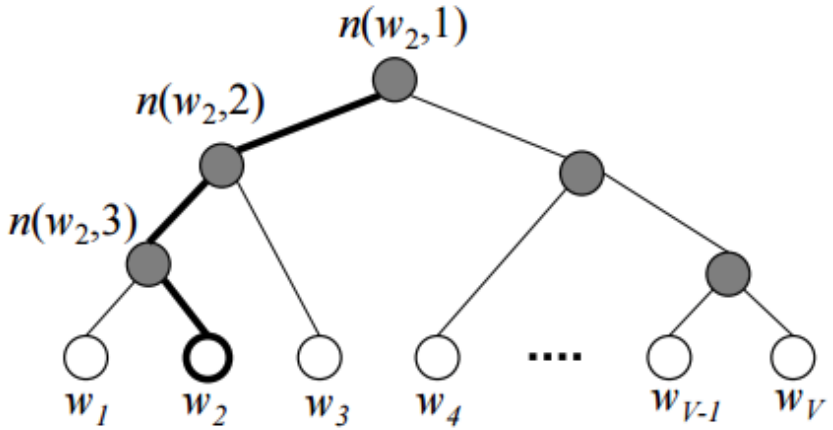
1. Иерархический softmax
2. Negative Sampling

Иерархческий Softmax

Над словами строится бинарное дерево. Вероятность i-го определяется как вероятность попасть в i-ый лист дерева из корня:

Иерархческий Softmax

Над словами строится бинарное дерево. Вероятность i-го определяется как вероятность попасть в i-ый лист дерева из корня:



$$p(w|w_I) = \prod_{i}^{L(w)-1} \sigma\left(\left[n(w,j+1) = ch\left(n(w,j)\right)\right] \cdot {v'_{n(w,j)}}^T v_I\right),$$

где $v'_{n(w,j)}$ – дополнительные параметры модели.

Теперь функция потерь вычисляется за $O(\log|\Omega|)$

Negative Sampling

- Наша функция потерь поощряет одно значение быть 1, а все остальные – 0.
- Изменим функцию потерь так, чтобы градиент считался только по тому значению, которое ожидается быть близким к 1 и $q \in [5, 20]$ другим значениям.

$$\log p(w_O|w_I) \longleftarrow \log \sigma({v'_{w_O}}^T v_{w_I}) + \sum_{i=1}^{q} \mathbb{E}_{w_i \sim P(w)} \left[\sigma(-{v'_{w_i}}^T v_{w_I}) \right]$$

$P(w)$ – считаем, используя счётчики в степени $3/4$

Subsampling. Результаты

- Частотные слова не несут в себе большой смысловой нагрузки, но при этом значительно увеличивают размер выборки.
- Будем удалять слова из обучающей выборки с вероятностью: $p(w) = 1 - \sqrt{\frac{t}{\text{freq}(w)}}$, $t = 10^{-5}$

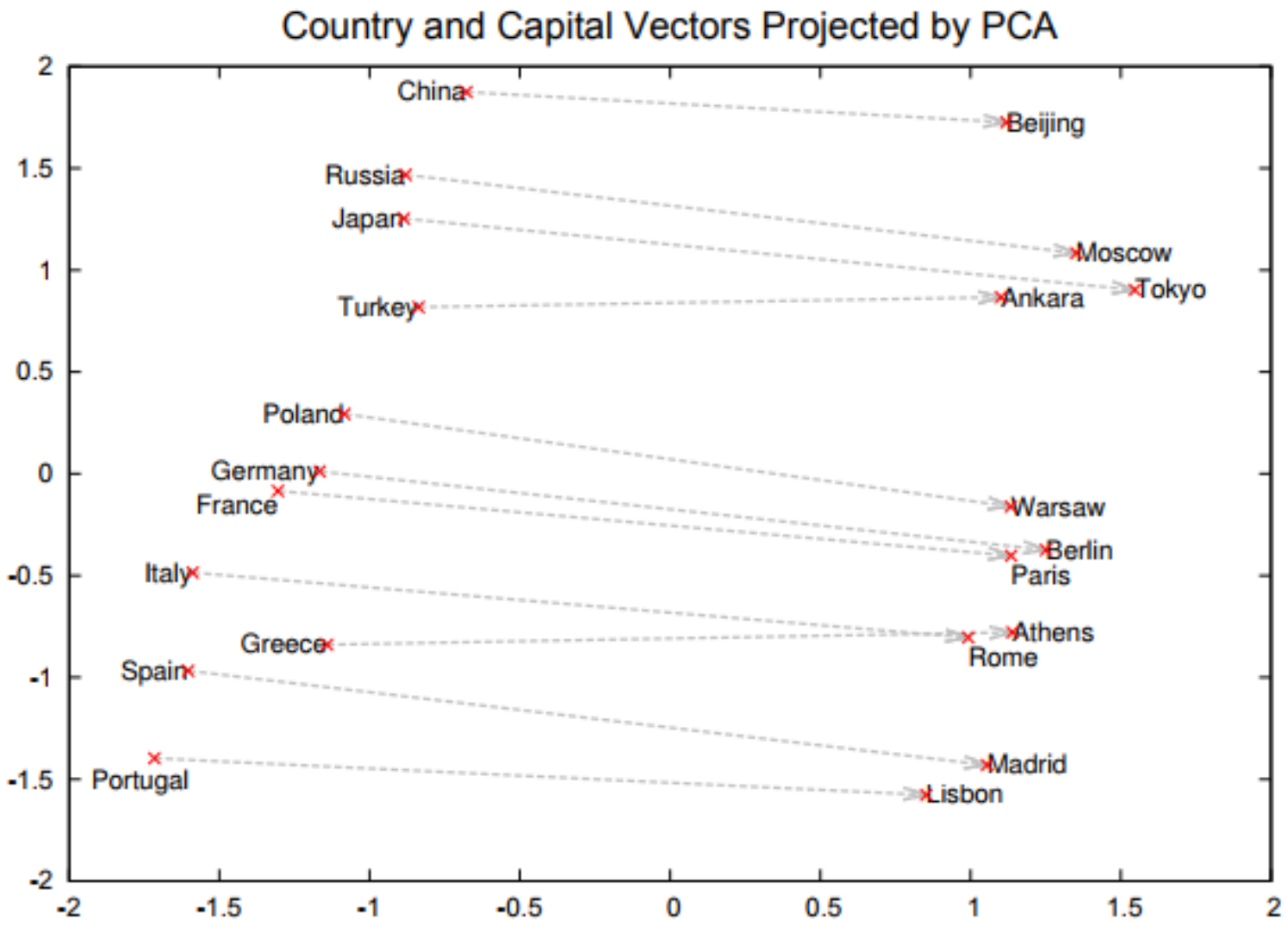
Subsampling. Результаты

- Частотные слова не несут в себе большой смысловой нагрузки, но при этом значительно увеличивают размер выборки.
- Будем удалять слова из обучающей выборки с вероятностью: $p(w) = 1 - \sqrt{\frac{t}{\text{freq}(w)}}$, $t = 10^{-5}$

Method	Dimensionality	No subsampling [%]	10^{-5} subsampling [%]
NEG-5	300	24	27
NEG-15	300	27	42
HS-Huffman	300	19	**47**

- Negative Sampling работает быстрее, чем Hierarchical Softmax, но хуже работает с менее частотными словами.
- Стандартный выбор: skip-gram + negative sampling + окно (5-10) + Subsampling + ($|h| = 300$)

Результаты


Country and Capital Vectors Projected by PCA
China
Beijing
Russia
Moscow
Japan
Tokyo
Turkey
Ankara
Poland
Warsaw
Germany
Berlin
France
Paris
Italy
Athens
Greece
Rome
Spain
Madrid
Portugal
Lisbon
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2