Векторное представление слов

Елена Тутубалина

Казанский федеральный университет

29 марта 2018 г.

Введение

- Векторное представление слова (word embedding) - вещественный вектор в пространстве с фиксированной невысокой размерностью
- Вход - коллекция текстов
- Выход - векторные представления слов из словаря коллекции

Области применения

- Поиск синонимов
- Исправление опечаток
- В качестве признаков для моделей
- Тематическое моделирование
- Средство изучения языка

Основня идея

Слова в похожих контекстах близки по смыслу:

- Он приоткрыл ИКС и заглянул в комнату
- Входная ИКС была не заперта
- Он постучал в ИКС и вошел

Какое слово скрыто под ИКС?

Определение похожести

- Со-встречаемости первого порядка (**syntagmatic association**).

  Слова рядом в тексте: 'выпил' и 'чай', 'включил' и 'телевизор'
- Со-встречаемости второго порядка. (**paradigmatic association**)

  У слов похожие соседи: 'преподаватель' и 'учитель', 'лампа' и 'светильник'

Дистрибутивная гипотеза

Счетчики совместной встречаемости слов

- Скользящее окно фиксированной ширины
- positive Pointwise Mutual Information

$$PMI = log \frac{p(u, v)}{p(u)p(v)} = log \frac{n_{uv} n}{n_u n_v}$$

Идея латентного семантического анализа (LSA)

- По корпусу текстов D со словарём T строим матрицусо-встречаемостей $X_{|T|\times|T|}$.
- Понижаем размерность, используя *сингулярное разложение матриц* (SVD).

$$X \approx UV^T$$

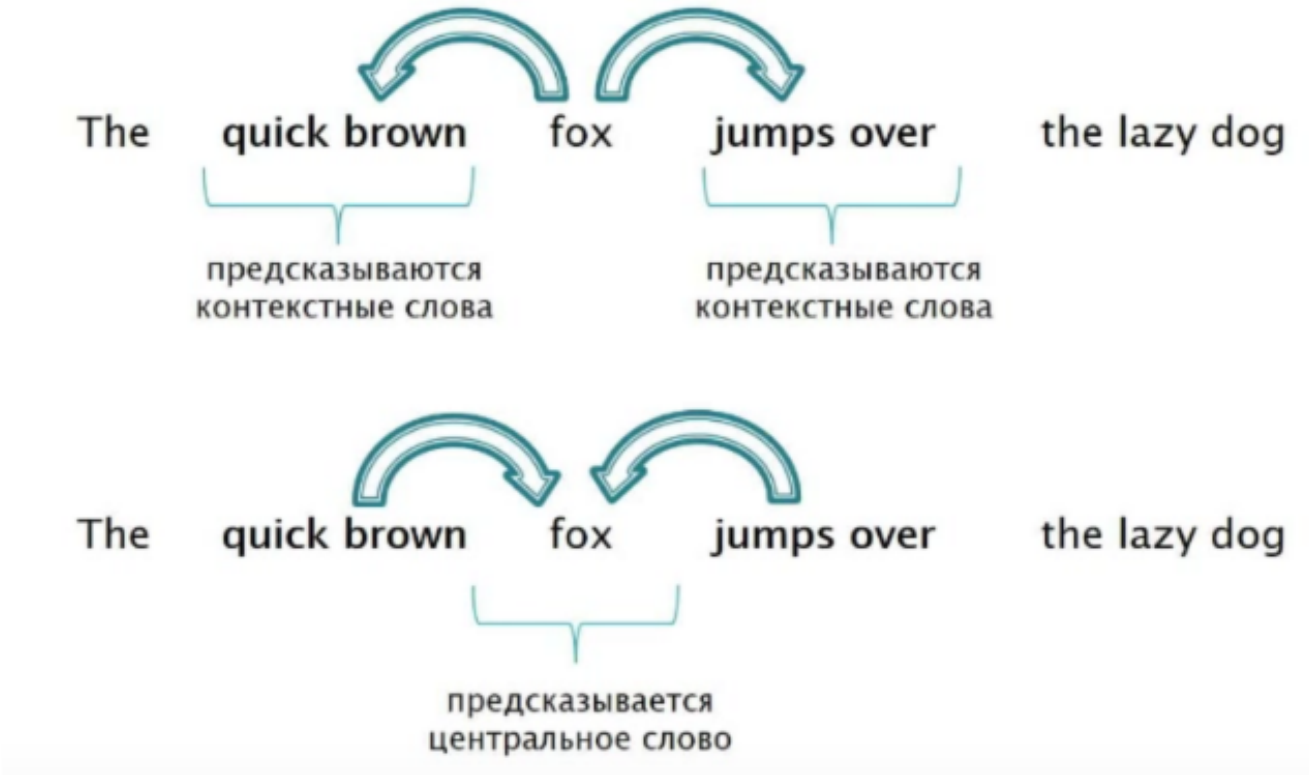
Недостатки LSA

- Относительно низкое качество получаемых представлений.
- Сложность работы с очень большой и разреженной матрицей
- Сложность добавления новых слов/документов.

Модель word2vec

word2vec — группа алгоритмов для получения векторных представлений слов.

Две модели: Continuous BOW и Skip-gram.



[1]

[1] https://www.youtube.com/watch?v=jvGbFS5AYOY

Модель CBOW

Логарифм правдоподобия: $L = \sum_{w \in D} \log p(w|c, \theta)$

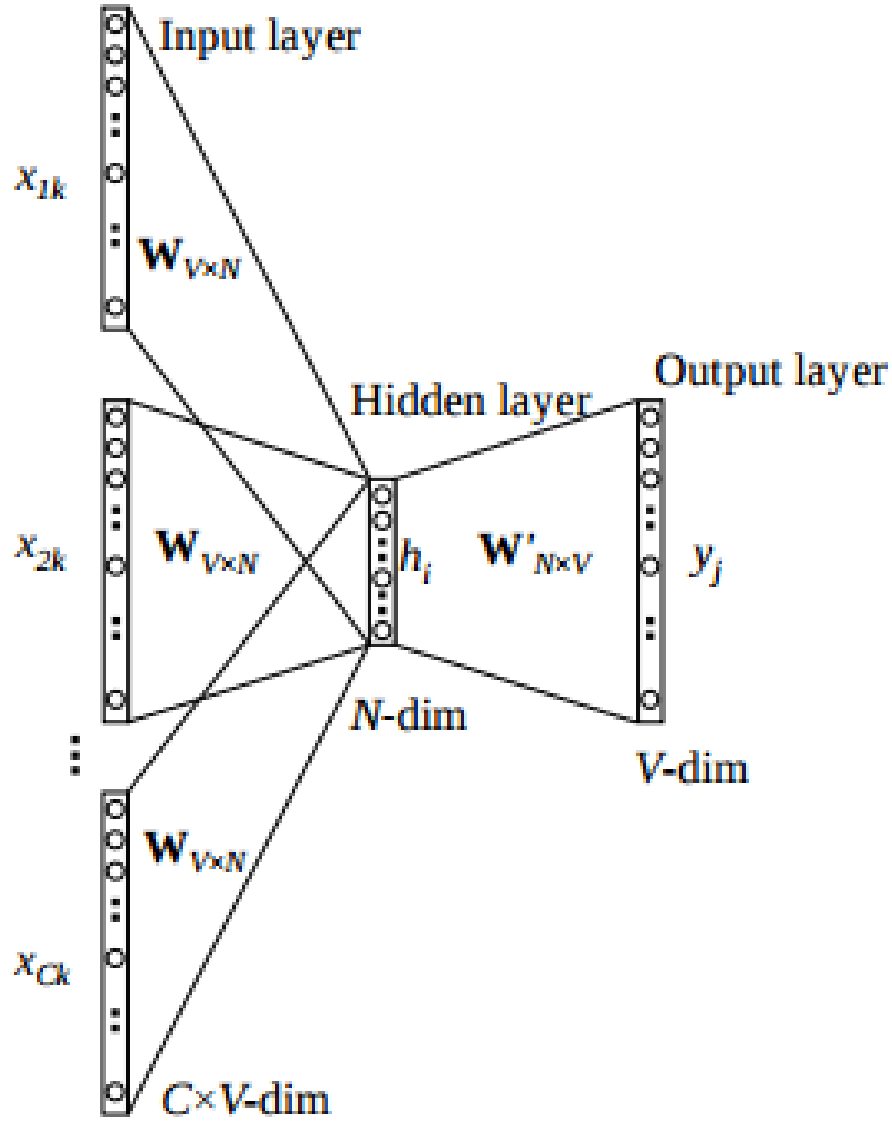
- θ - параметр
- w - текущее слово
- c - контекст слова

Обучаем с помощью простой нейросети

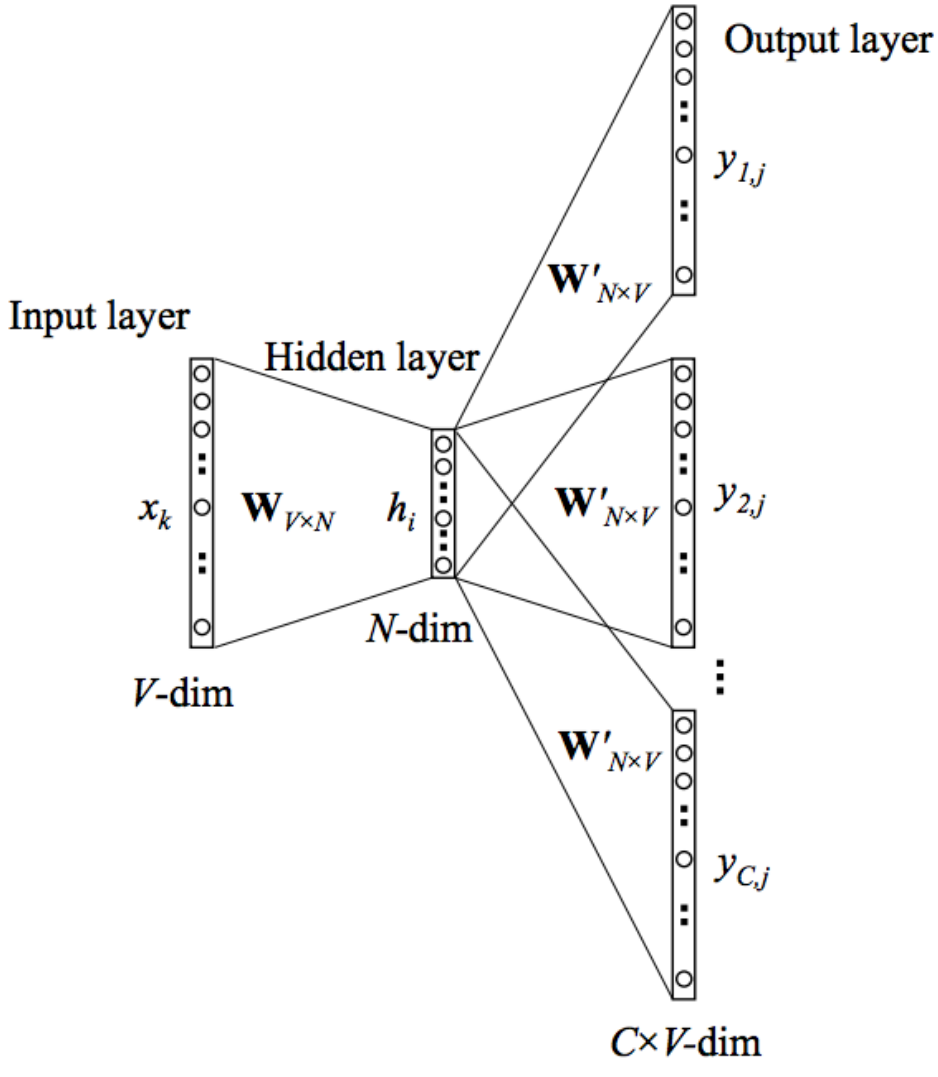
- Скользящим окном проходим по всей коллекции
- Вход - one-hot представление слова, вектор длины $|T|$
- Выход - распределение на словах коллекции, вектор длины $|T|$
- Вероятность $p(w|c, \theta)$ моделируется softmax-функцией.

$$p(w|c, v_w, v_c) = \frac{\exp(v_w^T, v_c)}{\sum_{w \in D} \exp(v_w^T, v_c)}$$

Модель CBOW


Input layer
Output layer
Hidden layer
x_{1k}
x_{2k}
x_{Ck}
$\mathbf{W}_{V\times N}$
$\mathbf{W}_{V\times N}$
$\mathbf{W}_{V\times N}$
h_i
$\mathbf{W}'_{N\times V}$
y_j
N-dim
V-dim
$C\times V$-dim

Модель Skip-gram



Negative sampling

- Подсчёт нормировочной константы в softmax — дорогая операция.
- Можно изменить постановку задачи и функционал качества.
- Решаем задачу бинарной классификации:

$$z = 1, if pair(w, s) \in D, z = 0 - else(s \in c(w))$$

$$p(z = 1|(w, s)) = \frac{1}{1 + \exp(-v_w^T, v_s)} = \sigma(v_w^T, v_s)$$

- Новый функционал правдоподобия:

$$L = \sum_{(w,s)\in D_1} \log \sigma(v_w^T, v_s) + \sum_{(w,s)\in D_2} \log \sigma(-v_w^T, v_s)$$

$$D_1 = (w, s) : s \in c(w), D_2 = (w, s) : s \notin c(w)$$

Negative sampling

$$L = \sum_{(w,s)\in D_1} \log \sigma(v_w^T, v_s) + \sum_{(w,s)\in D_2} \log \sigma(-v_w^T, v_s)$$

- В качестве отрицательных примеров для каждого рассматриваемого слова w генерируются случайные слова из T
- Функционал оптимизируется с помощью SGD.

Реализации

- Оригинальный word2vec
- Medallia/Word2VecJava
- FastText
- Spark MLLib Word2Vec
- Gensim word2vec
- и другие

Gensim — пакет для тематического моделирования, включает ряд полезных инструментов (часто в качестве удобной обёртки над готовыми реалищациями). Предоставляет интерфейс для работы с оригинальным word2vec.

Задание

- На тренировочном корпусе текстов из StackOverflow обучить модель word2vec
- Вывести 10 слов наиболее похожих на слова: Android, Java, program
- Добавить полученное векторное распределение слов в качестве признака классификатору
- Вывести метрики оценки качества классификации: полнота, точность, F-мера, сравнить с показателями без word2vec