



Автоматическая обработка текстов на естественном языке. Метод кодирования слов word2vec.

Евгений Борисов

NLP word2vec

Способ описание текста

частотный анализ

- нужен достаточный размер текста
- не учитывает последовательность

NLP word2vec

Способ описание текста

частотный анализ

- нужен достаточный размер текста
- не учитывает последовательность

кодирование отдельных слов

- можно использовать для коротких сообщений
- можно учитывать последовательность

NLP word2vec

способ кодирования слов

тривиальный способ

составить словарь, отсортировать и занумеровать

NLP word2vec

способ кодирования слов

тривиальный способ

составить словарь, отсортировать и занумеровать

Недостатки: номер не отражает смысла

NLP word2vec

PMI pointwise mutual information

оценка совместного использования слов u v

$$PMI(u, v) = \log \left(\frac{p(u, v)}{p(v) p(u)} \right)$$

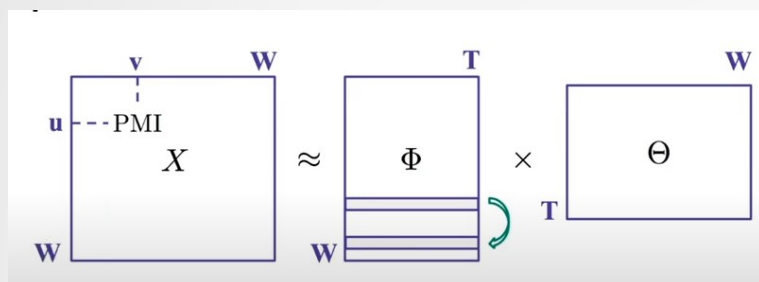
$p(u, v)$ – частота использования словосочетания

$p(u)$ и $p(v)$ - частота использования слов

NLP word2vec

используем контекст для кодирования слов

применим матричное разложение к квадратной матрице PMI



для кодирования слов используем матрицу Φ

NLP word2vec

способ кодирования слов Word2Vec

построим модель ML и обучим её кодировать слова по контексту

из текста извлекаем словарь W

каждому слову из W ставим в соответствие точку из V

$$w \mapsto v: W \rightarrow V; V \subset \mathbb{R}^n$$

NLP word2vec

способ кодирования слов Word2Vec

построим модель ML и обучим её кодировать слова по контексту

из текста извлекаем словарь W

каждому слову из W ставим в соответствие точку из V

$$w2v: W \rightarrow V; V \subset \mathbb{R}^n$$

совместно употребляемые в тексте слова из W

отображаются в близкие точки пространства V

$$w2v[\text{king}] - w2v[\text{man}] + w2v[\text{woman}] \approx w2v[\text{queen}]$$

NLP word2vec

Как это работает?

подготовка данных Word2Vec – учитываем контекст слов.

- очищаем текст T от лишних символов
- из очищенного текста T собираем словарь W
- для каждого слова w собираем контекст (окрестность)
т.е. слова удалённые от w не более чем на s позиций в T
- выполняем унитарное кодирование(one-hot encoding) W

$P_i:$
0 0 1 0 0

$Q_i:$
0 1 0 0 0
0 0 0 0 1
0 0 0 1 0
1 0 0 0 0
0 0 0 1 0

NLP word2vec

Как это работает?

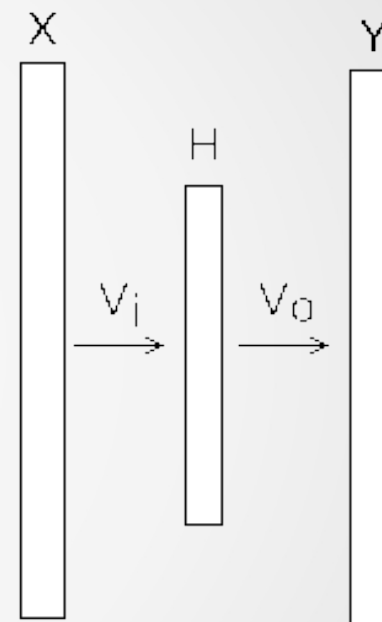
нейросеть Word2Vec

размер входного слоя X = размеру словаря W
= размеру выходного слоя Y

скрытый слой H - линейная активация

выходной слой Y — активация softmax

$$Y = \text{softmax}((X \cdot V_i) \cdot V_o)$$



конечный результат - матрица
внутренних представлений V_i

NLP word2vec

обучение сети word2vec

метод градиентного спуска

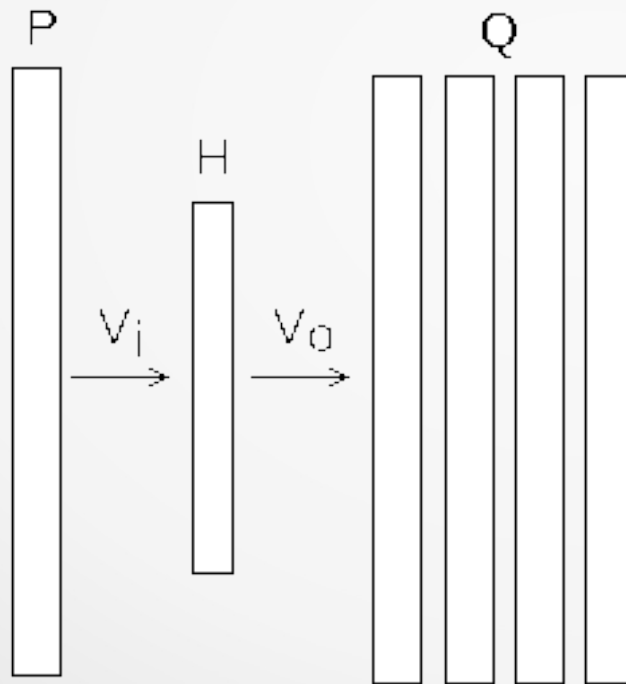
одна из двух стратегии

- Skip-Gram - по слову восстанавливаем контекст.
- CBOW(Continuous Bag of Words) по контексту восстанавливаем слово

NLP word2vec

обучение сети word2vec

- Skip-Gram - по слову восстанавливаем контекст.



NLP word2vec

обучение сети word2vec - Skip-Gram - по слову восстанавливаем контекст.

1. на вход сети подаётся код слова P ,
вычисляем состояние скрытого слоя H
вычисляем выход сети O

2. вычисляем значение функции потерь

если значение потерь увеличилось
то конец работы

$$E_i = \left| \log \sum \exp(U_i) - \sum_j (U_i * Q_{ij}) \right|$$

3. для каждого слова контекста Q_j и входа P :

вычисляем ошибку D на выходе сети O
и изменение весов сети ΔV_o , ΔV_i

$$\begin{aligned} D &= O - Q_j \\ \Delta V_{oj} &= H^T \cdot D \\ \Delta V_{ij} &= D^T \cdot P \cdot V_o^T \end{aligned}$$

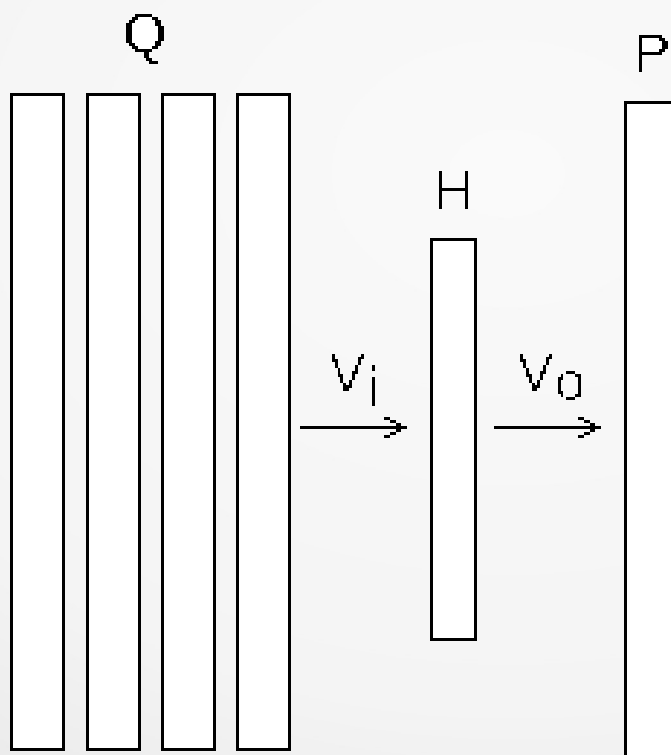
4. вычисляем суммарное изменение
весов сети ΔV_o , ΔV_i
корректируем веса
и повторяем цикл для другого слова P

$$\begin{aligned} \Delta V_o &= \sum_j \Delta V_{oj} \\ \Delta V_i &= \sum_j \Delta V_{ij} \end{aligned}$$

NLP word2vec

обучение сети word2vec

- CBOW(Continuous Bag of Words) по контексту восстанавливаем слово



NLP word2vec

обучение сети word2vec - CBOW, по контексту восстанавливаем слово

1. на вход сети подаётся усреднённое значение контекста Q ,
вычисляем состояние скрытого слоя H
вычисляем выход сети O

$$H = \frac{1}{c} \sum_{j=1}^c Q_j \cdot V_i$$

$$U = H \cdot V_o$$
$$O = \text{softmax}(U)$$

2. вычисляем значение функции потерь

если значение потерь увеличилось
то конец работы

$$E_i = \left| \log \sum \exp(U_i) - \sum (U_i * P_i) \right|$$

3. для каждого слова контекста Q_j и кода слова P ,
вычисляем ошибку D на выходе сети O
и изменение весов сети ΔV_o , ΔV_i .

$$D = O - P$$
$$\Delta V_o = H^T \cdot D$$
$$\Delta V_i = \sum_j D^T \cdot Q_j \cdot V_o^T$$

4. корректируем веса
и повторяем цикл для другого слова P

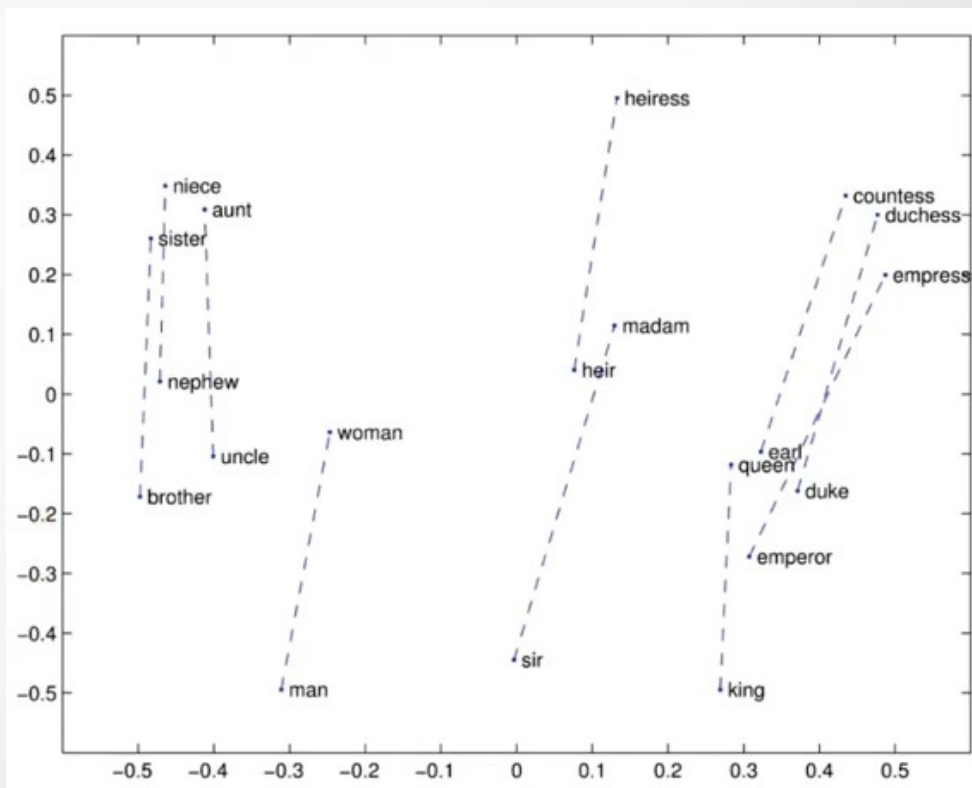
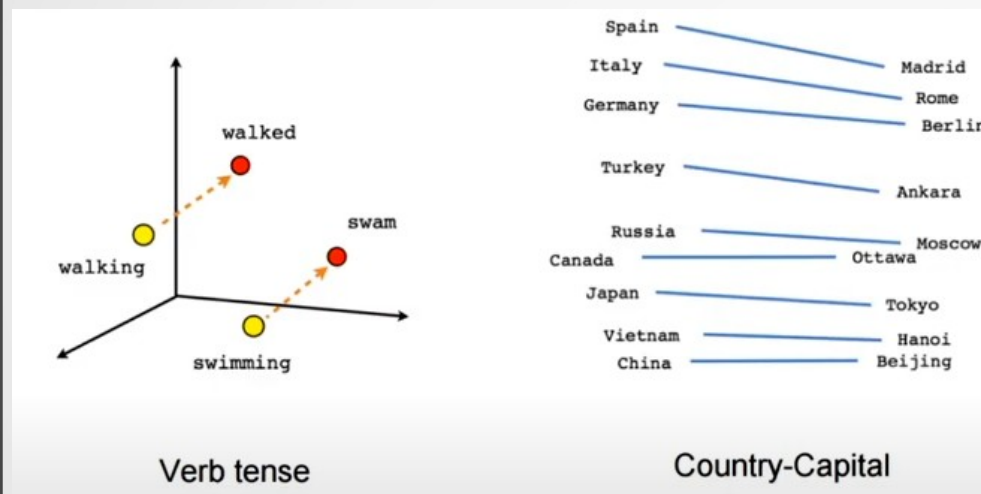
NLP word2vec

Результат работы

```
{ 'состоится': ['конечно', 'бал'],  
  'волнения': ['описывать', 'виргинский'],  
  'печатных': ['листа', 'липутину'],  
  'лямшин': ['вечер', 'чуткости'],  
  'или': ['отчасти', 'нет'],  
  'крышу': ['под', 'надлежащего'],  
  'общее': ['пригорюнясь', 'пострадать'],  
  'степанович': ['осилил', 'новых'],  
  'часов': ['кучке', 'часу'],  
  'вас': ['для', 'завтра'],  
  'слишком': ['милости', 'была'],  
  'крикнул': ['попа', 'прочтет'],  
  'видеть': ['свое', 'будто'],  
  'наготове': ['кармазинове', 'беспокойством'],  
  'помещался': ['этою', 'подыматься'],  
  'выполнено': ['планете', 'несмотря'],  
  'тебя': ['люблю', 'взрывов'],  
  'началом': ['фунтов', 'центру'],  
  'страдал': ['заявите', 'тесно'],  
  'отчаянная': ['новая', 'кончив']}
```

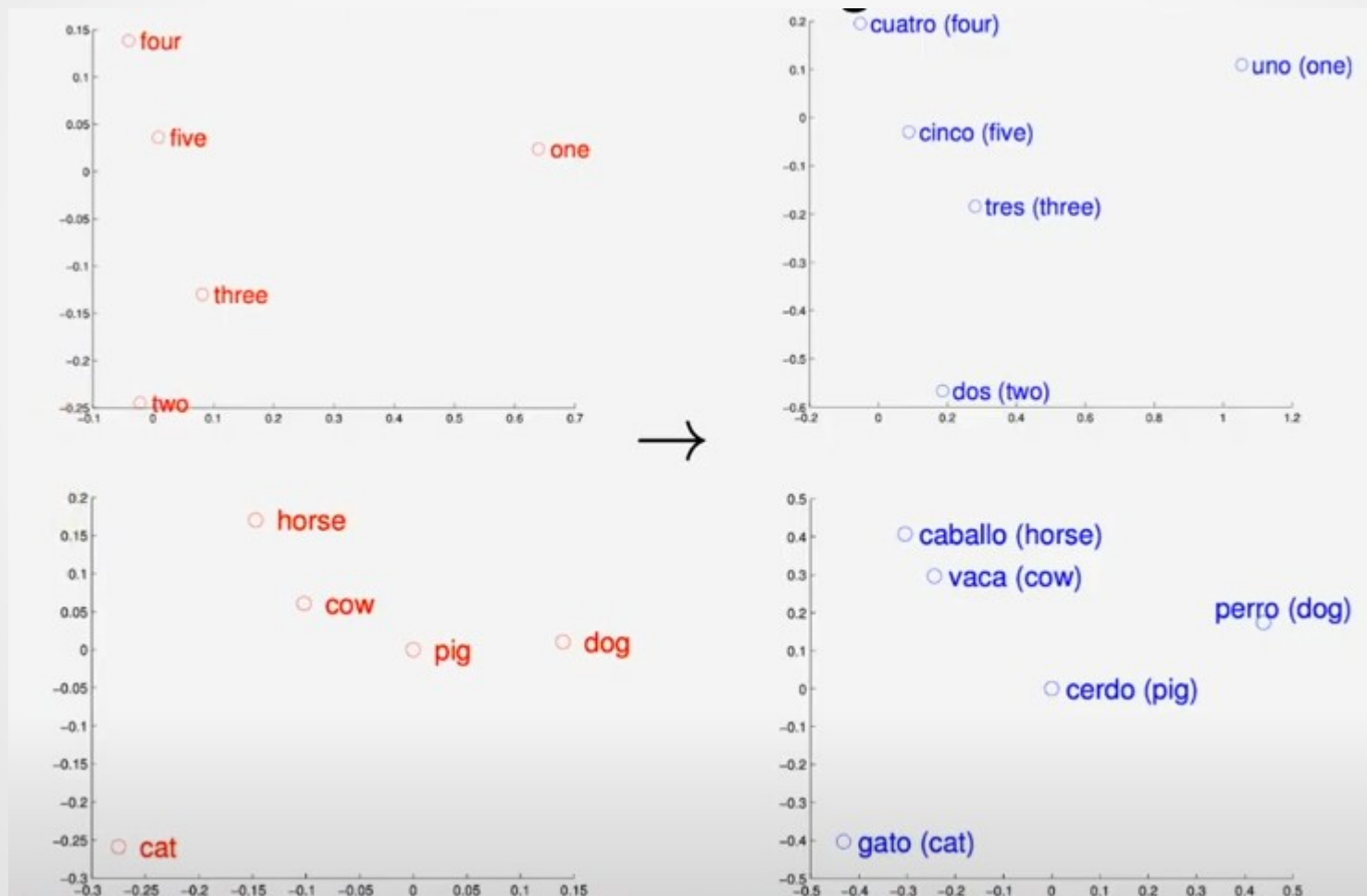
NLP word2vec

близкие по контексту слова отображаются в близкие точки w2v



NLP word2vec

взаимное расположения w2v в разных языках похожи

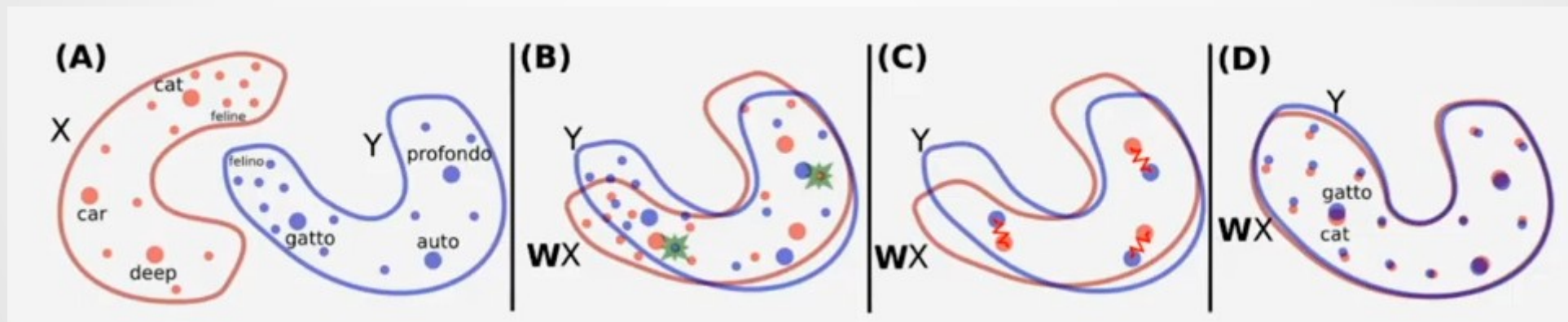


NLP word2vec

взаимное расположения $w2v$ в разных языках похожи

зная перевод некоторых слов и на основе этого построив отображение из $w2v$ пространства одного языка в другое,

мы получаем перевод всех остальных слов на основе контекста



NLP word2vec

косинусная мера близости – оценка сонаправленности w2v

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}}$$

NLP word2vec

Литература

`git clone https://github.com/mechanoid5/ml_nlp`

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, Jeffrey Dean
Distributed Representations of Words and Phrases and their
Compositionality

Радослав Нейчев Прикладное машинное обучение 1.Intro to NLP. Word
embeddings - Лекторий ФПМИ

Николай Карпачев Прикладное машинное обучение. Семинар 1.Word embeddings
- Лекторий ФПМИ

Евгений Борисов 0 методе кодирования слов word2vec
<http://mechanoid.su/ml-w2v.html>