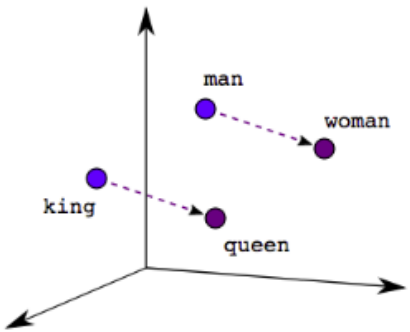
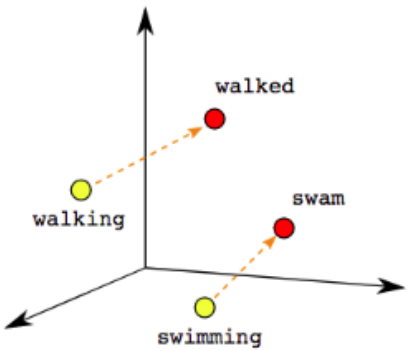
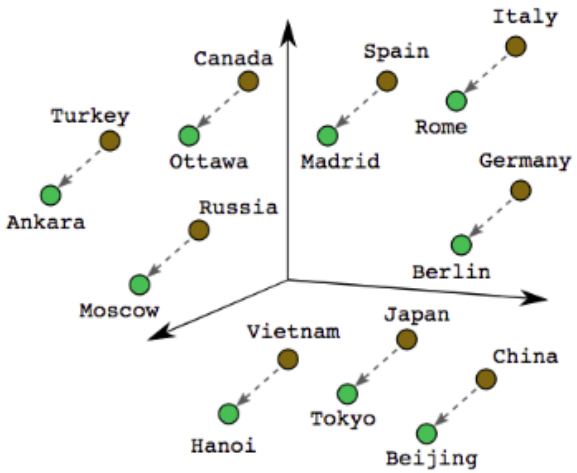
Извлечение эмбеддингов объектов

Виктор Китов
victorkitov.github.io







Содержание

Стандартное представление слов

- Обозначим D=размер словаря.
- Стандартные представления слов $x \in \mathbb{R}^D$:
  - $x_w = \mathbb{I}[w$ встретился в документе$]$
  - $x_w = TF_w = \#[w$ встретился в документе$]$
  - $x_w = TF_w IDF_w$, $IDF_w = \frac{N}{N_w}$
    - N - # документов
    - N_w - # документов, содержащих w хотя бы раз.
- V велико, поэтому нужно компактное представление (word embedding) $x \in \mathbb{R}^K$, $K << D$:
  - меньше входов=>меньше параметров=>ниже переобучение
  - возможность учитывать семантическое сходство/различие
    - например, синонимы "автомобиль" и "машина"

Интерпретируемые векторные представления слов

- Можно из слов извлекать интерпретируемые признаки:
  - x^1: часть речи
  - x^2: род (м/ж/ср - для существительных)
  - x^3: время (пр/наст/буд - для глаголов)
  - x^4: $\mathbb{I}$[начинается с заглавной буквы]
  - x^5: # букв
  - x^6: категория: машинное обучение, физика, биология, ...
  - x^7: подкатегория: обучение с учителем, без учителя, частичное обучение, ...
  - ...
- Необходимо придумывать признаки под задачу, производить разметку.
- Легче работать с неинтерпретируемыми признаками, но которые извлекаются автоматически.

Неинтерпретируемые представления слов

- Хотим, чтобы семантически близким словам соответствовали близкие представления.
- Дистрибутивная гипотеза (distributional hypothesis): слова близки по смыслу <=> они часто встречаются совместно
- "точность бустинга", "бустинг дал точность", "ниже точность, по сравнению с бустингом"
  - "точность" и "бустинг" связаны!
- Типичная размерность векторного представления $\in [300, 500]$.

Другие представления текста

Обрабатывать текст можно

- **на уровне символов**

Другие представления текста

Обрабатывать текст можно

- **на уровне символов** - их мало, но обработка медленная

Другие представления текста

Обрабатывать текст можно

- **на уровне символов** - их мало, но обработка медленная
- **на уровне n-грамм символов**:
  - биграммы для «*Мы идем*»: *Мы, ы_ , _и, ид, дё, ём, м.*
  - хорошо для новых слов и слов с опечатками

Другие представления текста

Обрабатывать текст можно

- **на уровне символов** - их мало, но обработка медленная
- **на уровне n-грамм символов**:
  - биграммы для «*Мы идем*»: *Мы, ы_ , _и, ид, дё, ём, м.*
  - хорошо для новых слов и слов с опечатками
- **на уровне фраз**: неслучайно часто встречающиеся слова (коллокации) склеиваются.

$$(w_i, w_j)\text{-коллокация} \iff \frac{p(w_i w_j) - \delta}{p(w_i)p(w_j)} > threshold$$

  - $\delta \in [0, 1]$ - параметр, снижающий значимость редко совстречающихся слов.

Содержание

Матрица совстречаемости слов

- $C \in \mathbb{R}^{DxD}$ - матрица со-встречаемости слов.
- c_{ij} =#{слово j встретилось в контексте слова i}.
- Пример для контекста ± 1 слово:

I like deep learning. I like NLP. I enjoy flying.

counts	I	like	enjoy	deep	learning	NLP	flying	.
I	0	2	1	0	0	0	0	0
like	2	0	0	1	0	1	0	0
enjoy	1	0	0	0	0	0	1	0
deep	0	1	0	0	1	0	0	0
learning	0	0	0	1	0	0	0	1
NLP	0	1	0	0	0	0	0	1
flying	0	0	1	0	0	0	0	1
.	0	0	0	0	1	1	1	0

Разложение матрицы совстречаемости

- Hyperspace Analogue to Language (HAL)[1]: эмбеддинги извлекаются
  - как строки U
  - либо столбцы V^T в сокращённом сингулярном разложении:

$$C = U\Sigma V^T$$

[1] Lund and Burgess, 1996.
[2] Bullinaria and Levy, 2007

Разложение матрицы совстречаемости

- Hyperspace Analogue to Language (HAL)[1]: эмбеддинги извлекаются
  - как строки U
  - либо столбцы V^T в сокращённом сингулярном разложении:

$$C = U\Sigma V^T$$

- Проблема: эмбеддинги доминируются частыми словами! Решение:

$$c_{ij} \to \log(c_{ij}+1)$$

- Дальнейшее улучшение: $c_{ij} \to PPMI(w_i, w_j)$[2]

$$PPMI(w_i, w_j) = \max\{0, PMI(w_i, w_j)\} = \max\{0, \ln \frac{P(w_i, w_j)}{P(w_i)P(w_j)}\}$$

[1] Lund and Burgess, 1996.
[2] Bullinaria and Levy, 2007

GloVe

- GloVe[3]: матр. факторизация $\log C \approx P^T \widetilde{P} + B + \tilde{B}$

$$\sum_{i,j=1}^{D} f\left(c_{ij}\right)\left(\mathbf{p}_i^T \tilde{\mathbf{p}}_j + b_i + \tilde{b}_j - \log c_{ij}\right)^2 \to \min_{\left\{\mathbf{w}_i, \tilde{\mathbf{w}}_i, b_i, \tilde{b}_i\right\}_{i=1,2,\ldots D}}$$

$f(c_{ij})$ - веса ($\uparrow$ функция от c_{ij} , $f(0)=0$)

[3] https://aclanthology.org/D14-1162.pdf

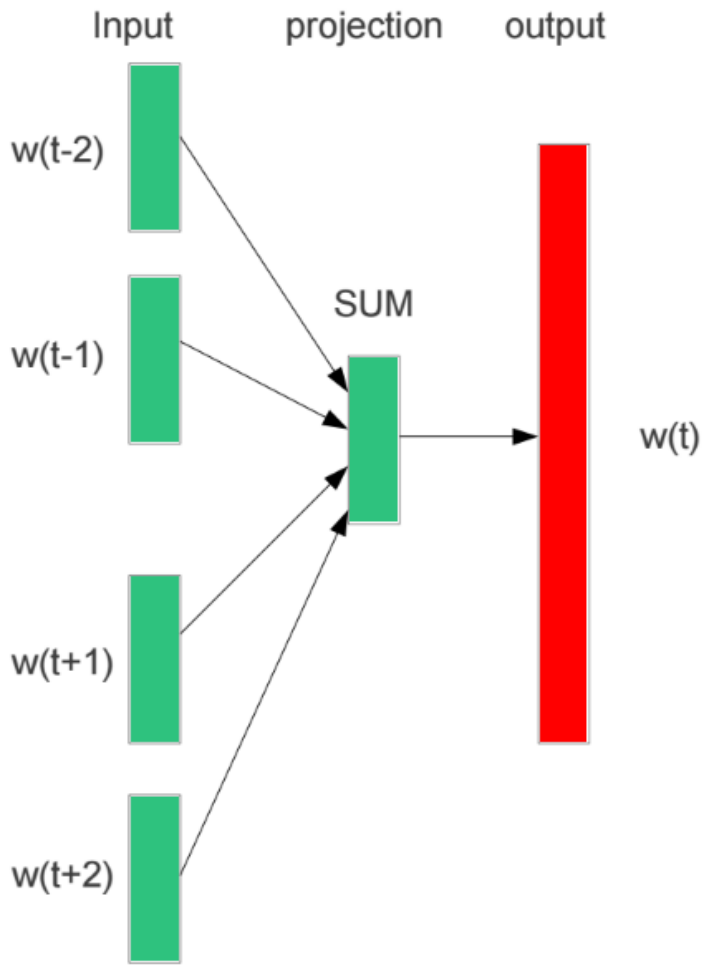
Содержание

Word2vec

- Для каждого w оценим:
  - целевое представление слова $\mathbf{v}_w$
  - контекстное представление слова $\mathbf{u}_w$
    - впоследствии можно не использовать, усреднить или конкатенировать с целевым представлением

CBOW: идея

Continuous bag of words (CBOW): предсказываем центральное слово по контексту.



CBOW: модель

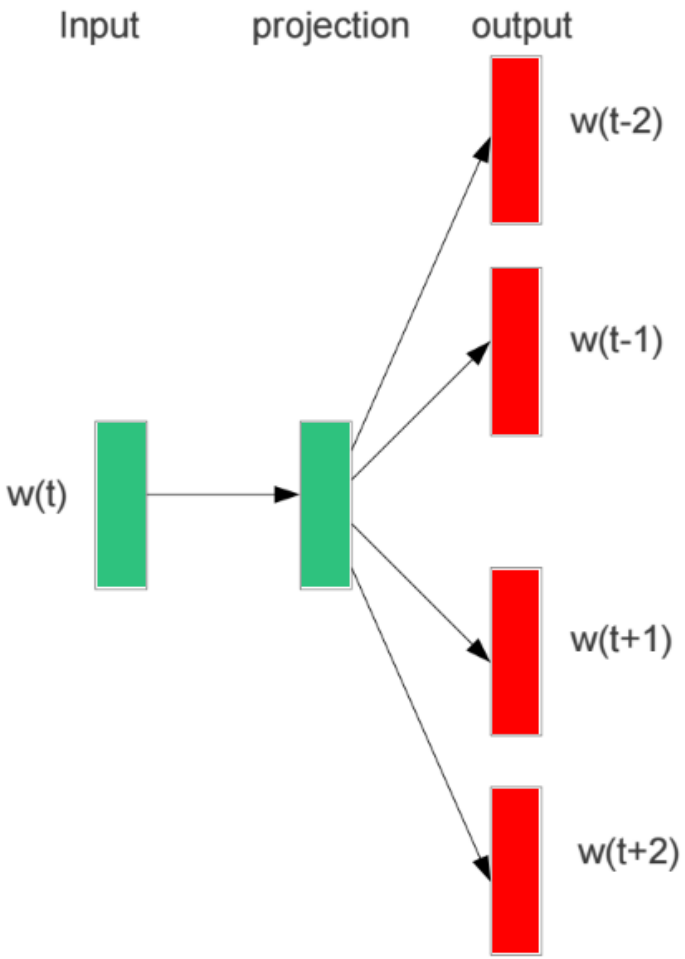
$$\frac{1}{T}\sum_{t=1}^{T}\ln p(w_t|w_{t-K},..w_{t-1},w_{t+1},...w_{t+K}) \to \max_{\{\mathbf{u}_w,\mathbf{v}_w\}_w}$$

где $\mathbf{u}_c = \sum_{-K\le i\le K, i\neq 0}\mathbf{u}_{w_{t+i}}$ и

$$p(w_t|w_{t-c},..w_{t-1},w_{t+1},...w_{t+c}) = \frac{\exp\left(\mathbf{u}_c^T\mathbf{v}_{w_t}\right)}{\sum_{w=1}^{D}\exp\left(\mathbf{u}_c^T\mathbf{v}_w\right)}$$

Skip-gram: идея

Skip-gram: предсказываем контекст по центральному слову:



Skip-gram: модель

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-K\le i\le K,\, i\neq 0}\ln p(w_{t+i}|w_t)\to\max_{\{\mathbf{u}_w,\mathbf{v}_w\}_w}$$

$$p(w_{t+i}|w_t)=\frac{\exp\left(\mathbf{u}_{w_t}^T\mathbf{v}_{w_{t+i}}\right)}{\sum_{w=1}^{D}\exp\left(\mathbf{u}_{w_t}^T\mathbf{v}_w\right)}$$

Комментарии

- Можем использовать ансамбли представлений
  - сумма, среднее, конкатенация
- Можем извлекать представления и др. объектов из последовательностей:
  - страницы, посещённые в ходе веб-сессии
  - сервисы, заказанные клиентом у компании
  - нуклеотиды в ДНК последовательности

Содержание

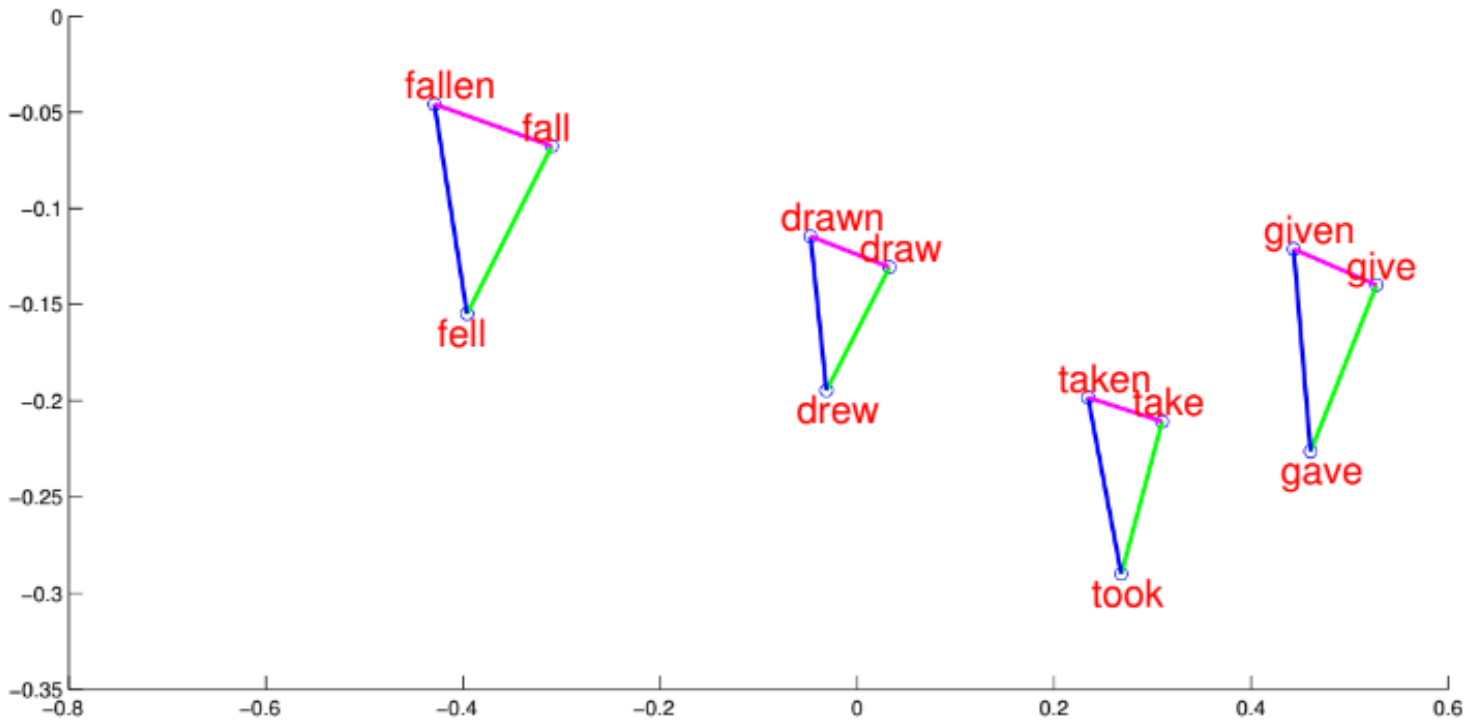
Похожие слова по представлению[4]

- Ближайшие соседи слова в пространстве эмбеддингов - слова, похожие по смыслу
(корпус GoogleNews, cosine-sim):
  - student -> teacher, faculty, school, university
  - car -> truck, jeep, vehicle
  - country -> nation, continent, region

[4] http://epsilon-it.utu.fi/wv_demo/

Формы слов

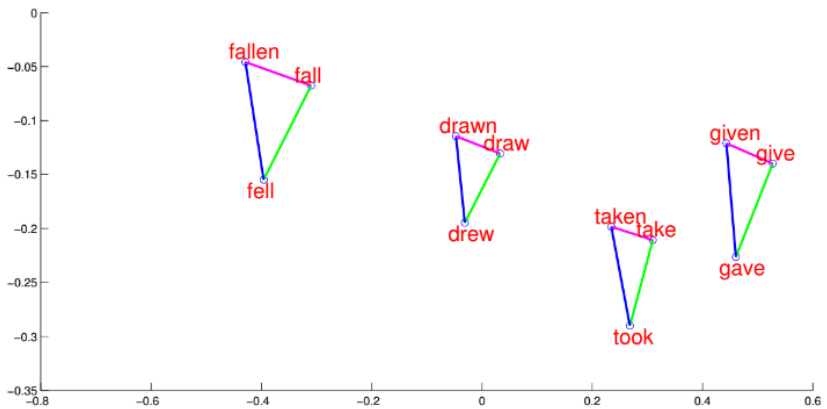
Одинаковые слова в разных формах образуют похожие структуры:



Представления могут помочь строить др. формы новых и редких слов.

Семантическая регулярность

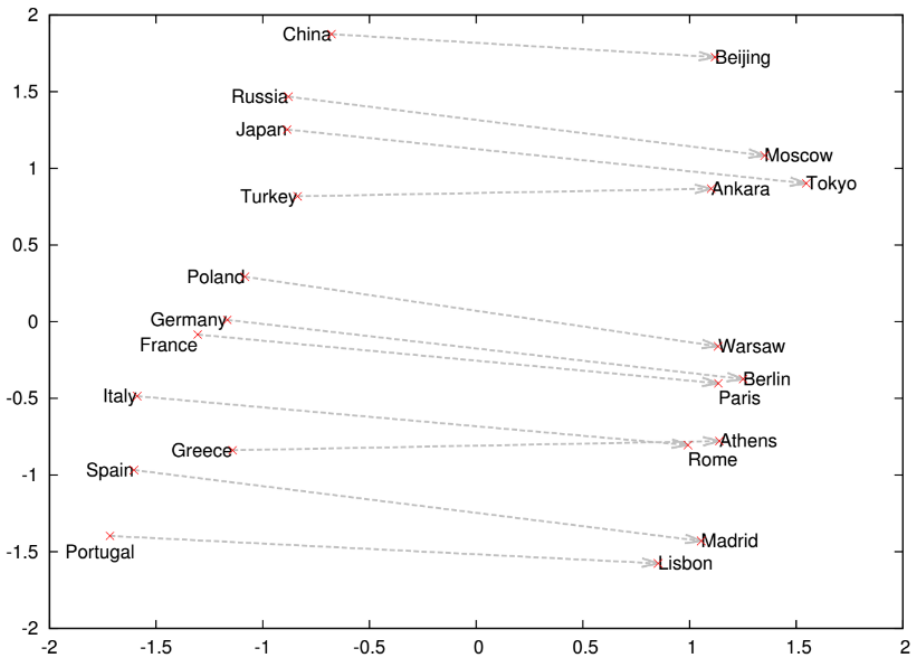
Слова, связанные семантически определенным образом группируются единообразно:



(prince-princess)+queen≈king. Может помочь в системе автоматических ответов на вопросы.

Семантическая регулярность

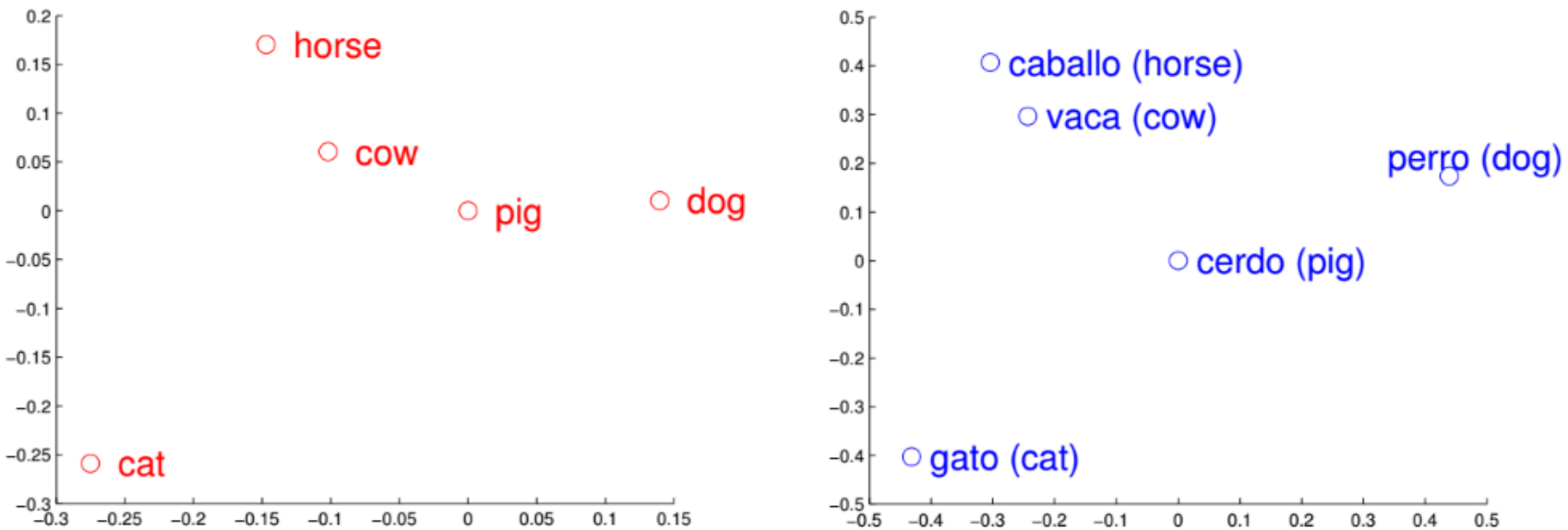
Слова, связанные семантически определенным образом группируются единообразно:



(Beijing-China)+Russia≈Moscow! Может помочь в системе автоматических ответов на вопросы.

Слова на разных языках

Слова на разных языках группируются похожим образом:



Достаточно простого линейного преобразования, чтобы примерно отображать слова на похожих языках друг в друга.

Содержание

Вычислительная сложность Word2vec

CBOW: $$\frac{1}{T}\sum_{t=1}^{T}\ln p(w_t|w_{t-c},..w_{t-1},w_{t+1},...w_{t+c}) \to \max_{\{\mathbf{u}_w,\mathbf{v}_w\}}$$

где $u_c = \sum_{-K\le i\le K, i\neq 0} u_{w_{t+i}}$ и

$$p(w_t|w_{t-K},..w_{t-1},w_{t+1},...w_{t+K}) = \frac{\exp\left(\mathbf{u}_c^T\mathbf{v}_{w_t}\right)}{\sum_{w=1}^{D}\exp\left(\mathbf{u}_c^T\mathbf{v}_w\right)}$$

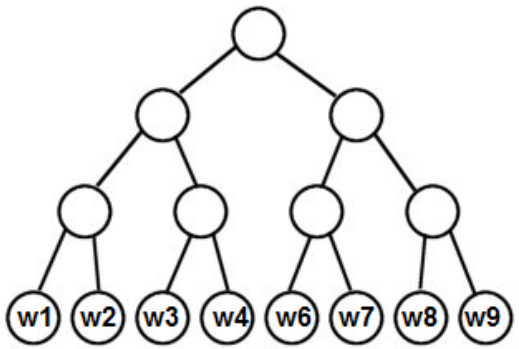
SkipGram: $$\frac{1}{T}\sum_{t=1}^{T}\sum_{-K\le i\le K, i\neq 0}\ln p(w_{t+i}|w_t) \to \max_{\{\mathbf{u}_w,\mathbf{v}_w\}}$$

$$p(w_{t+i}|w_t) = \frac{\exp\left(\mathbf{u}_{w_t}^T\mathbf{v}_{w_{t+i}}\right)}{\sum_{w=1}^{D}\exp\left(\mathbf{u}_{w_t}^T\mathbf{v}_w\right)}$$

Проблема: знаменатель вычисляется за $O(D)$.

Иерархический SoftMax

- Рассмотрим Skip-Gram.
- Построим бинарное дерево, где листья - предсказываемые слова.



- $p\left(w_O|w_I\right)$ посчитаем как путь в бинарном дереве, где
  - w_I - входное слово с эмбеддингом u_{w_I}.
  - w_O - выходное (предсказываемое) слово
  - $\mathbf{v}_w \to \{\mathbf{v}_j\}_j$- выходные эмбеддинги каждого внутр. узла j
- Переход из узла j к следующему узлу:

$$p(\text{left}|j) = \sigma\left(\mathbf{u}_c^T\mathbf{v}_j\right),$$
$$p(\text{right}|j) = 1-\sigma\left(\mathbf{u}_c^T\mathbf{v}_j\right) = \sigma\left(-\mathbf{u}_c^T\mathbf{v}_j\right)$$

- $p\left(w_O|w_I\right)$ =произведение вероятностей до листа w_I.

Иерархический SoftMax

- Сложность вычисления $p(w_O|w_I)$ существенно снижается:

$$O(D) \to O(\log_2 D)$$

- Целевым представлением для w теперь уже является входной эмбеддинг u_w
  - а не набор выходных, связанных с переходами в дереве
- Эффективнее работает не сбалансированное дерево, а дерево Хафмана
  - более частотным словам - более короткие пути.

Негативное сэмплирование

- Негативное сэмплирование (negative sampling)[5] - аппроксимация максимизация правдоподобия.
- Для каждой реальной (позитивной) пары (w_t, w_{t+i}) в SkipGram сэмплируем K негативных случайно $(w_t, w_{j_1}), \dots (w_t, w_{j_K})$.

$$\underbrace{\ln\left(\frac{1}{1+e^{-u_{w_t}^T v_{w_{t+i}}}}\right)}_{\sigma\left(+u_{w_t}^T v_{w_{t+i}}\right)} + \sum_{k=1}^{K} \underbrace{\ln\left(\frac{1}{1+e^{+u_{w_t}^T v_{w_{j_k}}}}\right)}_{\sigma\left(-u_{w_t}^T v_{w_{j(k)}}\right)} \to \max_{\{u_w, v_w\}_w}$$

- S ~2-5. $p\left(w_{j(k)}\right) \propto p(w)^{3/4}$- чаще сэмлируем редкие слова.
  - позволяет точнее оценивать их эмбеддинги

[5] Distributed Representations of Words and Phrases

fastText[6]

- В классическом skip-gram совместимость была $u_t^T v_{t+i}$.
- В fastText $w \to \left[u_w,\ \{p_j\}_{p\in\text{n-grams}(w)},\ v_w\right]$
  - $u_w, \{p_j\}_{p\in\text{n-grams}(w)}$ - входные эмбеддинги для известного слова
  - v_w - выходной эмбеддинг для предсказываемого
  - 3-граммы для <person>: <pe, per , ers , rso , son, on>.
  - использовались все n-граммы, $3 \le n \le 6$.
- Новая совместимость

$$u_{w_t}^T v_{w_{t+i}} + \sum_{j\in\text{n-grams}(w_t)} p_j^T v_{w_{t+i}}$$

- Для w вне словаря $\to$ только n-граммы $\sum_{j\in\text{n-grams}(w)} p_j$.

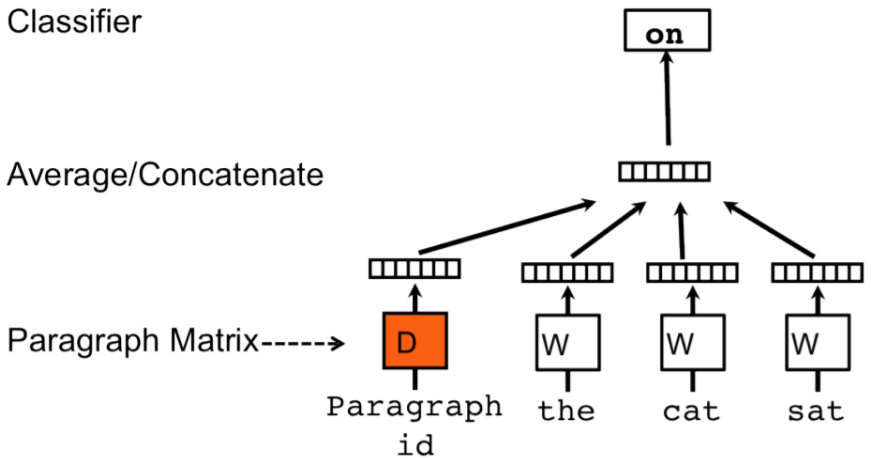
[6] https://arxiv.org/pdf/1607.04606.pdf, код и данные:fasttext.cc

Содержание

Представления параграфов - мотивация

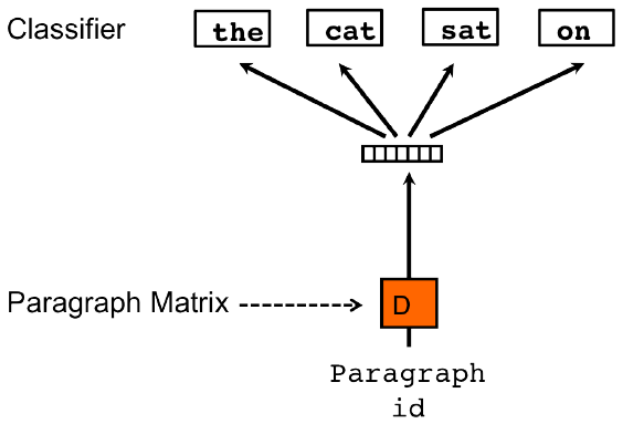
- Необходимо получить векторные представления параграфов (документов, предложений,...).
- Простой подход: усреднить слова, входящие в параграф.
  - или взвешенно усреднить, учитывая частоту встречаемости слов и их тематику.
- Точнее работает непосредственное представление самих параграфов.

Paragraph vector: модель PV-DM



- Во время обучения делим документы на параграфы. Каждому параграфу -> векторное представление.
- Оценивается CBOW, контекст: представление слов и парграфов.
- Можно усреднять или конкатенировать контексты слов и параграфа.
- Называется *Distributed Memory Model of Paragraph Vectors (PV-DM)*.

Paragraph vector: модель PV-DBOW

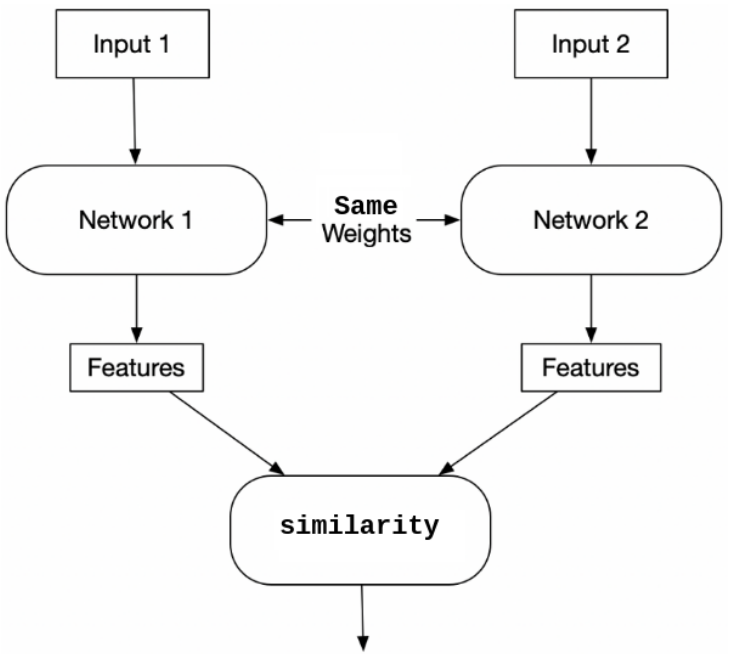


- Во время обучения делим документы на параграфы. Каждому параграфу -> векторное представление.
- Оценивается skip-gram: предсказываются случайные слова параграфа по представлению параграфа.
  - контекст: только представления параграфов
- Называется *Distributed Bag of Words version of Paragraph Vector (PV-DBOW)*

Содержание

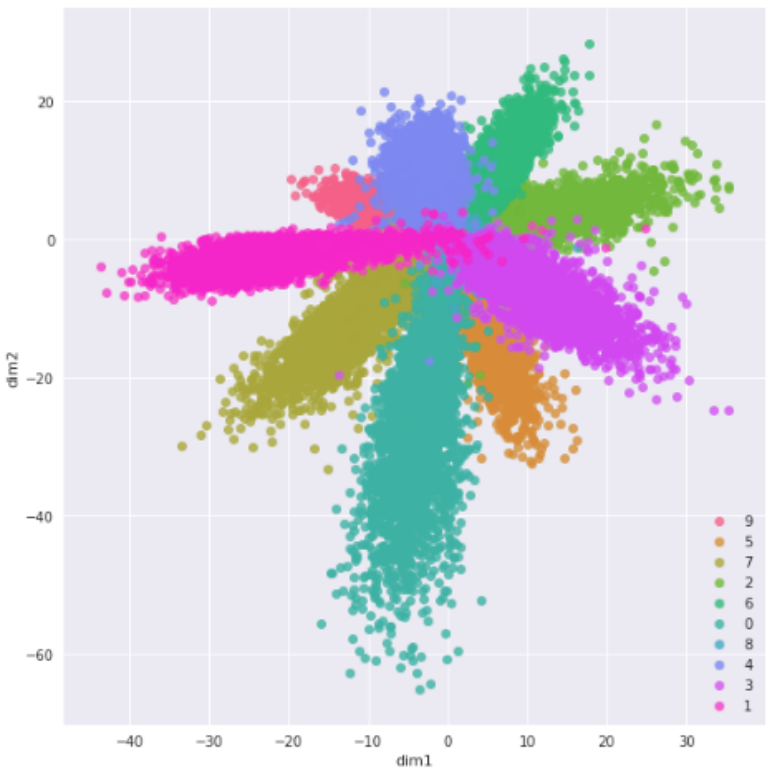
Сиамская сеть

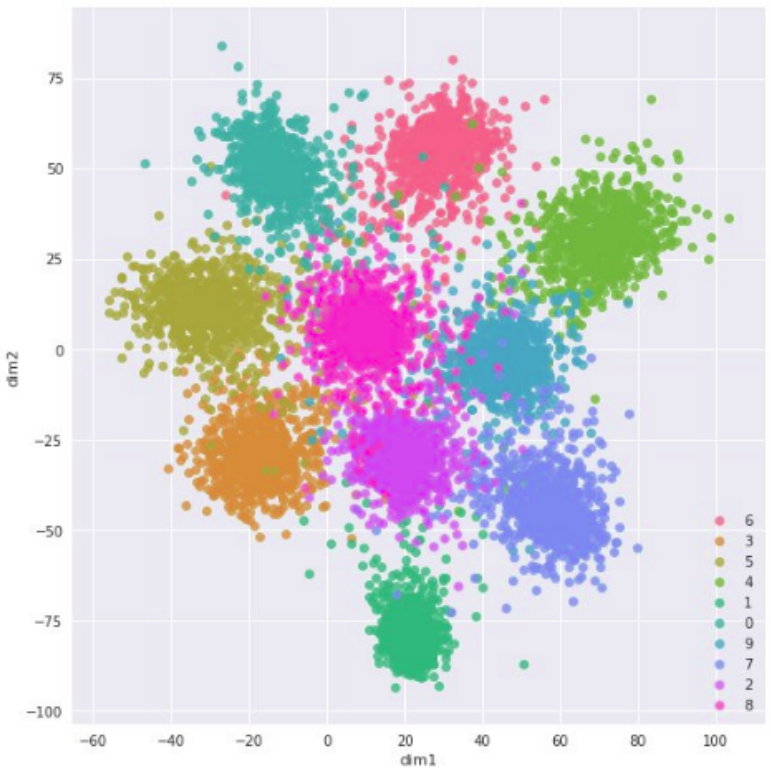
- **Сиамская сеть** (siamese net): объект $x_n \to$ эмбеддинг e_n.
- **Контрастное обучение**:
  - похожие объекты $\Rightarrow$ похожие представления
  - непохожие объекты $\Rightarrow$ непохожие представления
  - для классификации похожесть $\Leftrightarrow$ совпадение классов
  - похожесть: $\langle \cdot, \cdot \rangle$, cos-sim, $-\|\cdot\|_2^2$



Эмбеддинги $e(x)$: классификатор vs. сиамская сеть

Представления объектов внутри обычного классификатора и после сиамской сети для MNIST:





В пространстве эмбеддингов классификацию легко проводить $\forall$ метрическим методом!

Примеры приложений (одинаковый тип объектов)

- **Классификация:**
  - похожие: объекты одного класса
  - непохожие: объекты разных классов
- **Обнаружение перефразирования:**
  - объекты: фразы текста
  - похожие: фразы об одном и том же
  - непохожие: фразы о разном
- **Проверка подписи:**
  - объекты: сканы подписей
  - похожие: подписи одного человека
  - непохожие: подписи разных людей

Примеры приложений (разный тип объектов)

- **Рекомендательная система:**
  - объекты: пользователи, товары
  - похожие: пользователь и лайкнутый им товар
  - непохожие: пользователь и дизлайкнутый им товар
- **Поисковая система:**
  - объекты: запросы, документы
  - похожие объекты: запрос и релевантный документ
  - похожие : запрос и нерелевантный документ
  - возможен поиск по изображениям, видео, звукам!

Примеры приложений (разный тип объектов)

- **Рекомендательная система:**
  - объекты: пользователи, товары
  - похожие: пользователь и лайкнутый им товар
  - непохожие: пользователь и дизлайкнутый им товар
- **Поисковая система:**
  - объекты: запросы, документы
  - похожие объекты: запрос и релевантный документ
  - похожие : запрос и нерелевантный документ
  - возможен поиск по изображениям, видео, звукам!

Объекты разных типов обрабатывают разные сиамские сети, но принцип контрастного обучения сохраняется.

Попарные потери

Попарные потери (pairwise contrastive loss, spring loss)[7]:

- обучение на случайных парах объектов x_i, x_j

$$\mathcal{L}(x_i, x_j) = \begin{cases} \rho(e_i, e_j)^2, & \text{если } x_i, x_j \text{ похожи} \\ \max\{0,\ \alpha - \rho(e_i, e_j)\}^2, & \text{если } x_i, x_j \text{ непохожи} \end{cases}$$

- $\alpha > 0$ - гиперпараметр (мин. расстояние для непохожих объектов, когда не будет штрафа)
- $\rho(e, e')$ - обычно Евклидово
- число уникальных пар - $O(N^2)$.

[7] Выгоднее позволять похожим объектам небольшую вариацию в эмбеддингах. Предложите соответствующее изменение.

Тройные потери

Тройные потери (triplet loss):

- обучение на случайных тройках x, x^+, x^-.
- x - опорный объект (anchor)
- x^+ - похожий на x (positive)
- x^- - не похожий на x (negative)
- $\alpha > 0$ - гиперпараметр (мин. разница расстояний без штрафа)
- $\rho(e, e')$ - обычно Евклидово

$$\mathcal{L}(x, x^+, x^-) = \max\left\{\rho(e, e^+)^2 - \rho(e, e^-)^2 + \alpha; 0\right\}$$

- число уникальных пар - $O(N^3)$.

Вероятностные потери

Вероятностные потери (InfoNCE loss, NCE=noise constrastive estimation):

- x - опорный объект (anchor)
- x^+ - похожий на x (positive)
- $x_1, ...x_S$ - набор непохожих на x объектов

$$\mathcal{L}(x, x^+, x_1^-, ...x_M^-) = -\ln \frac{e^{sim(e,e^+)}}{e^{sim(e,e^+)} + \sum_{m=1}^{S} e^{sim(e,e_m^-)}}$$

$$sim(e, e') = \frac{e^T e'}{||e|| \cdot ||e'||}$$

- $> O(N^3)$ уникальных сэмплов.

Комментарии

- Сэмплировать можно равномерно
  - по объектам (максимизируем микро-усредненные метрики per object)
  - по классам (максимизируем макро-усредненные метрики per class)
- Контрастное обучение можно использовать для metric learning $\rho_\theta(x, x')$.

Сиамская сеть и классификация

- **Обычный классификатор:**
  - выучивает "что представляет каждый класс".
  - выдает степени соответствия x каждому классу.
- **Сиамская сеть:**
  - выучивает "что отличает классы друг от друга".
  - выдает расстояния от x до каждого класса.
  - более устойчива к дисбалансу классов и редким классам (*one shot learning*)
    - при обучении каждый класс учитывается поровну
    - модель выучивает признаки, по которым можно судить о сходстве классов на частотных классах, потом сразу подхватывает их для редких.
  - извлекает больше информации из выборки
    - обучение не на объектах, а на парах и тройках объектов.
  - хороша в ансамбле с классификатором (↑разнообразие)

Заключение

- **Представления слов** отображают слова в компактные векторные представления.
  - может применяться
    - к биграммам, триграммам, коллокациям.
    - к символам - удобно для новых слов
    - к любым объектам из посл–тей (нуклеотиды в ДНК и др.)
- **Представления параграфов** отображают параграфы в векторные представления.
  - работают лучше, чем усреднение слов параграфа
- Представления можно находить для целевой или связанной задачи (language modeling, transfer learning)
- **Сиамская сеть** оценивает эмбеддинги объектов.
  - применения: классификация (особенно one shot learning), нахождение похожих изображений, информационный поиск, рекомендательные системы, ...