Дистрибутивная семантика

Векторное представление слова

Екатерина Черняк

Факультет компьютерных наук НИУ ВШЭ

March 29, 2018

1. Введение
2. Счетные и нейронные модели представления слова
   - Факторизация матрицы терм-контекст
   - Word2Vec
   - Word2Vec как факторизация матрицы sPMI
3. Использование представлений слова
4. Другие модели
   - Word2Vec-f
   - Doc2Vec
   - GloVe
   - FastText
   - StarSpace
   - AdaGram
5. Другое
   - Двуязычные представления слов
   - HistWords
   - Составления предметных словарей эмоционально-окрашенных слов

Представление слова

Word representation [TRB10]

A *word representation* is a mathematical object associated with each word, often a vector. Each dimension's value corresponds to a feature and might even have a semantic or grammatical interpretation, so we call it a word feature.

Word embedding

A *word embedding* is a vector in a low dimensional space, which represents each word.

Примеры (http://rusvectores.org, [KK16])

мороз_NOUN

Выберите модель:

☑ Новостной корпус ☐ Araneum fastText ☐ Araneum Maximum ☐ НКРЯ и Wikipedia ☐ НКРЯ

Показывать только:

○ Наречия ○ Прилагательные ○ Имена собственные ○ Глаголы ○ Существительные ◉ Все части речи ○ Часть речи запроса

Найти похожие слова!

Семантические аналоги для *мороз* (ALL)

Новостной корпус

1. холод 0.59 
2. стужа 0.59
3. жара 0.55 
4. двадцатиградусный 0.52
5. тридцатиградусный 0.51
6. трескучий 0.49
7. вьюга 0.48
8. градусный 0.48
9. морозец 0.47
10. мазай 0.46

Примеры (http://rusvectores.org, [KK16])

мороз_NOUN → зима_NOUN

жара_NOUN → ???

Новостной корпус

1. лето 0.44 
2. зной 0.41
3. весна 0.38 
4. малоснежный 0.38
5. осень 0.37 

Обозначения

- $w \in V_W$ – слова, всего слов $|V_W|$
- $c \in V_C$ – контексты, всего контекстов $|V_C|$
- $\#(w, c)$ – сколько раз слово w встретилось в контексте c
- $(w, c) \in D$ – наблюдаемые пары (слово, контекст), всего пар $|D|$
  $\sum_w \sum_c \#(w, c) = |D|$
- $\boldsymbol{E} \in \mathbb{R}^{|V_w| \times d}$ – матрица эмбеддингов
- d – размерность эмбеддинга, $d \ll |V_W|$

Дистрибутивная семантика

Смысл слова [L. Wittgenstein]

Die Bedeutung eines Wortes liegt in seinem Gebrauch.

Distributional hypothesis [J.R.Firth]

You shall know a word by the company it keeps!

Дистрибутивная семантика

Векторная модель: матрица слово-контекст M

$$M_{[i,j]} = f(w_i, c_j)$$

	c_1	c_2	...	$c_{\lvert V_C \rvert}$
w_1	f_{11}	f_{12}		$f_{1\lvert V_C \rvert}$
w_2	f_{21}	f_{22}		$f_{2\lvert V_C \rvert}$
...				
$w_{\lvert V_W \rvert}$	$f_{\lvert V_W \rvert 1}$	$f_{\lvert V_W \rvert 2}$		$f_{\lvert V_W \rvert \lvert V_C \rvert}$

Дистрибутивная семантика

Векторная модель: матрица слово-контекст $\boldsymbol{M} \in \mathbb{R}^{V_W \times V_C}$

$$\boldsymbol{M}_{[i,j]} = f(w_i, c_j)$$

Как определить $f(w_i, c_j)$?

- $\#(w, v)$
- $P(w, c) = \frac{\#(w,v)}{|D|}$
- $\text{PMI}(w, c) = \log \frac{\#(w,c)|D|}{\#(w)\#(c)}$
- $\text{PPMI}(w, c) = \max(\text{PMI}(w, c), 0)$

Как определить $f(w_i, c_j)$?

Векторная модель: матрица слово-контекст $M \in \mathbb{R}^{V_W \times V_C}$

$$M_{[i,j]} = f(w_i, c_j)$$

$\#(w, v)$ или $P(w, c) = \frac{\#(w,v)}{|D|}$

	the	a	...	cute
w_1	f_{11}	f_{12}		$f_{1\|V_C\|}$
cat	f_{21}	f_{22}		$f_{2\|V_C\|}$
...				
$w_{\|V_W\|}$	$f_{\|V_W\|1}$	$f_{\|V_W\|2}$		$f_{\|V_W\|\|V_C\|}$

Как определить $f(w_i, c_j)$?

Векторная модель: матрица слово-контекст M

$$M_{[i,j]} = f(w_i, c_j)$$

$$\text{PMI}(w, c) = \log \frac{\#(w,c)|D|}{\#(w)\#(c)}$$

	the	a	...	cute
w_1	f_{11}	f_{12}		$f_{1\|V_C\|}$
cat	f_{21}	f_{22}		$f_{2\|V_C\|}$
...				
$w_{\|V_W\|}$	$f_{\|V_W\|1}$	$f_{\|V_W\|2}$		$f_{\|V_W\|\|V_C\|}$

$$\text{PMI}(w, c) = |\#(w, c) = 0| = \log 0 = -\infty \Rightarrow$$
$$\text{PPMI}(w, c) = \max(\text{PMI}(w, c), 0)$$

Как определить w, c?

1. Слова w
   - все слова
   - только существительные
   - именованные сущности
2. Контексты c
   - документы, абзацы, предложения
   - слова в пределах окна ($\pm$ k слов слева и справа от w)
   - глаголы по связям nsubj, dobj, iobj

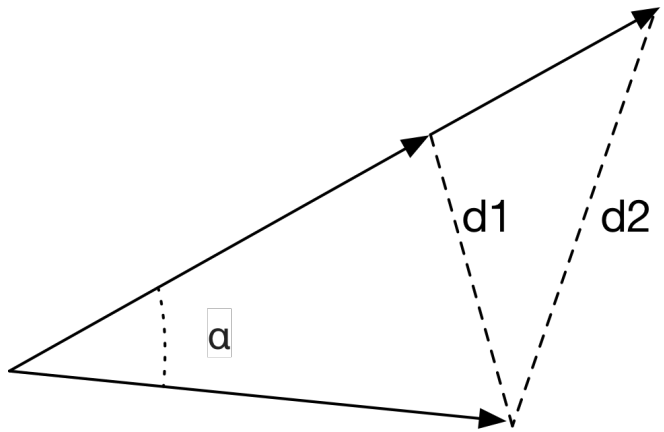
Оценка близости между словами

- Мера Жаккара:

$$jc(u,i) = \frac{\sum_i \min(u_i, v_i)}{\sum_i \max(u_i, v_i)}$$

- Евклидово расстояние:

$$d(u,v) = \sqrt{\sum_{i=1}^{n}(u_i - v_i)^2}$$



- Косинусная мера близости:

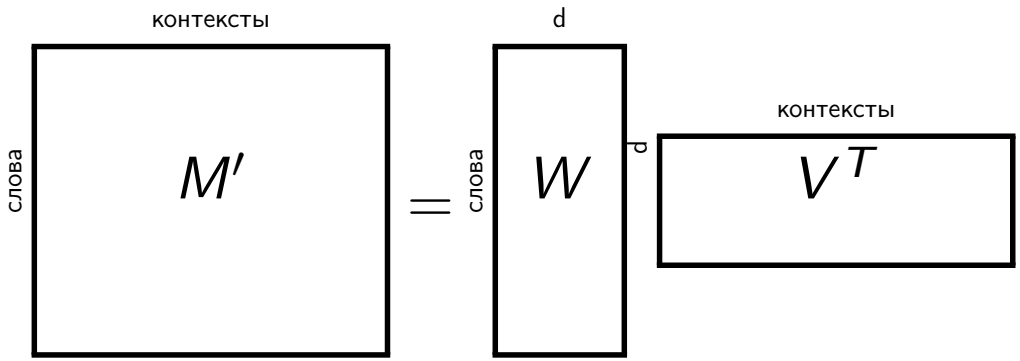
$$cos(u,v) = \frac{u \cdot v}{||u||_2||v||_2} = \frac{\sum_i u_i v_i}{\sqrt{\sum_i u_i^2}\sqrt{\sum_i v_i^2}}$$

Факторизация матрицы терм-контекст

Снижение размерности матрицы слово-контекст $\boldsymbol{M} \in \mathbb{R}^{V_W \times V_C}$:

$$M' = W \times V^\top, W \in \mathbb{R}^{V_W \times d}, V \in \mathbb{R}^{V_C \times V_d}$$

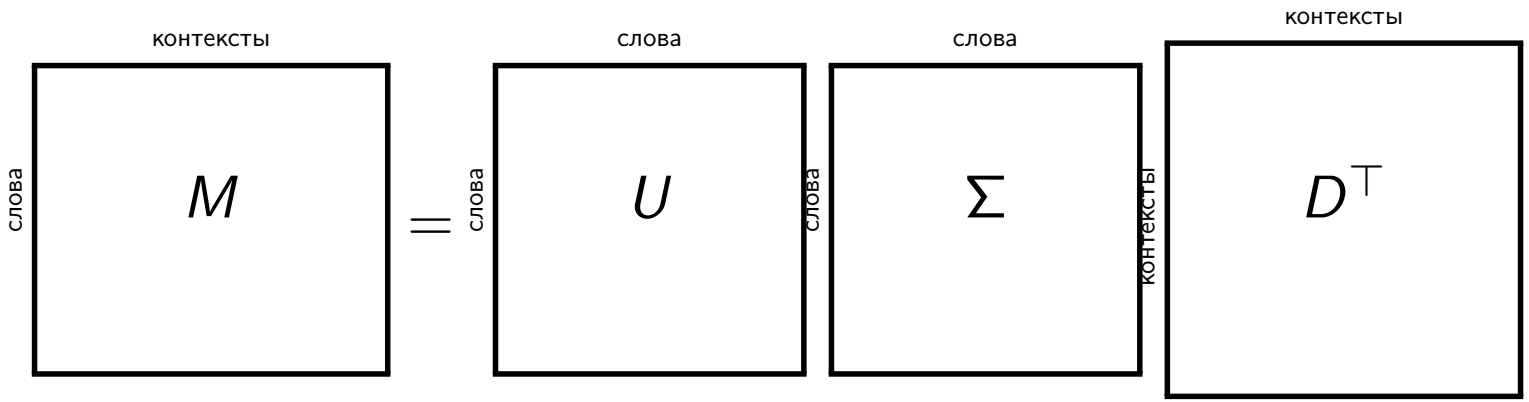
M' – лучшее приближение ранга d к M по L_2.



Факторизация матрицы терм-контекст

Сингулярное разложение матрицы слово-контекст $\boldsymbol{M} \in \mathbb{R}^{V_w \times V_c}$:

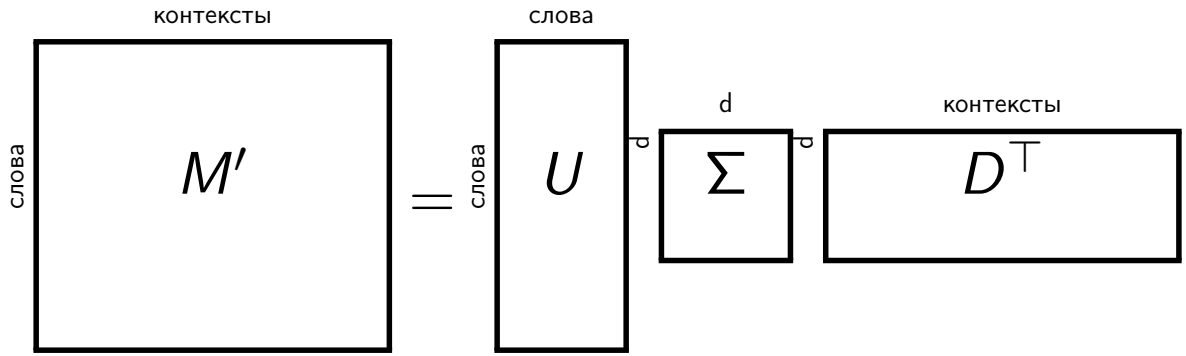
$$M = U\Sigma D^\top$$



Факторизация матрицы терм-контекст

Апроксимация ранга d матрицы слово-контекст $\boldsymbol{M} \in \mathbb{R}^{V_w \times V_c}$:

$$M'_d = U_d \Sigma_d D_d^\top$$



Искомое разложение M:

$$W = U_d \sqrt{\Sigma_d}, V^\top = \sqrt{\Sigma_d} D_d^\top$$

Примеры

- Туториал M. Baroni
- Курс A. Copestake и A. Herbelot

1. Введение
2. Счетные и нейронные модели представления слова
   - Факторизация матрицы терм-контекст
   - Word2Vec
   - Word2Vec как факторизация матрицы sPMI
3. Использование представлений слова
4. Другие модели
   - Word2Vec-f
   - Doc2Vec
   - GloVe
   - FastText
   - StarSpace
   - AdaGram
5. Другое
   - Двуязычные представления слов
   - HistWords
   - Составления предметных словарей эмоционально-окрашенных слов

SENNA [CW08]

Предсказываем слово w по его левому и правому контексту $c_{1:k}$:
$P(c_1 c_2 \dots w \dots c_{k-1} c_{k-2})$.
$v_w(w)$ – вектор слова w
$v_c(w)$ – вектора контекста c

$$s(w, c_{1:k}) = g(xU) \cdot v$$

$x = [v_c(c_1), \dots, v_c(c_k), v_w(w)]$, $U \in \mathbb{R}^{(k+1)d_{emb} \times d_h}$, $v \in \mathbb{R}^{d_h}$
Margin-based ranking loss:
$L(w, c, w') = \max(0, 1 - d(w, c_{1:k}) - s(w', c_{1:k}))$
Для каждой пары $(w, c_{1:k}$ сэмплируем случайное слово из словаря w'.

Word2Vec [MCCD13]

1. Две архитектуры:
   - Continuous bag-of-words model (CBOW)
   - skip-gram
2. Два критерия оптимизации:
   - Hierarchical softmax
   - Negative-sampling: для каждой пары $(w, c) \in D$ найти k слов, таких что $(w_k, c) \in \bar{D}$

D – множество наблюдаемых пар слово-контекст

$\bar{D}$ – множество ненаблюдаемых пар слово-контекст

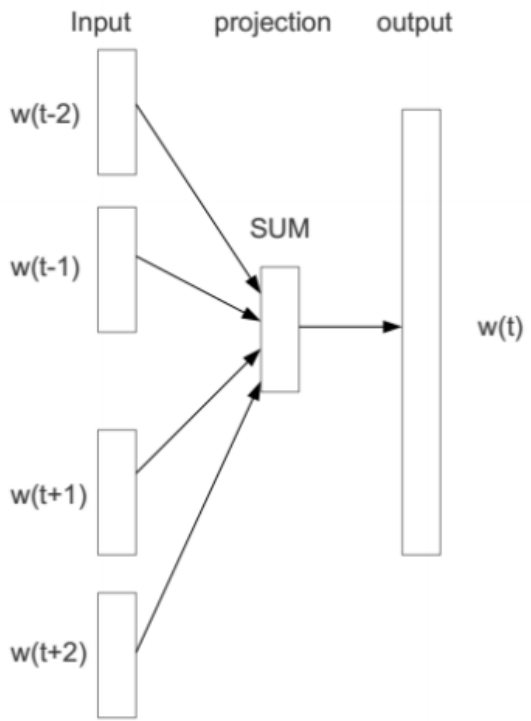
Вероятность $(w, c) \in D$: $P(D = 1|w, c) = \frac{1}{1+e^{-s(w,c)}}$

Оптимизационная задача:

$$L(\Theta, D, \bar{D}) = \sum_{(w,c)\in D} P(D = 1|w, c) + \sum_{(w,c)\in \bar{D}} P(D = 0|w, c)$$

Continuous bag-of-words model (CBOW) [MCCD13]

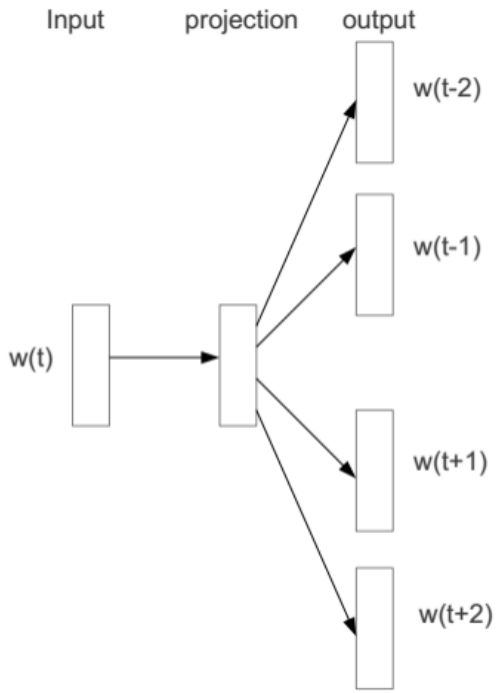
- Входной слой: контекст слова ($+, -\frac{k}{2}$ слова слева и справа)
- Слой проекции: линейный
- Выходной слой: вектор слова



$$P(D = 1|w, c_{1:k}) = \frac{1}{1 + e^{-(w \cdot c_1 + w \cdot c_2 + \ldots + w \cdot c_k)}}, c = \sum_{i=1}^{k} c_i$$

skip-gram [MCCD13]

- Обратная задача: предсказание векторов контекста по данному слову
- Выходной слой: вектор слов
- Все контексты независимы: $(w, c_1), \ldots, (w, c_k)$



$$P(D = 1|w, c_i) = \frac{1}{1 + e^{-(w \cdot c_i)}}$$

$$P(D = 1|w, c_{1:k}) = \prod_{i=1}^{k} P(D = 1|w, c_i) = \prod_{i=1}^{k} \frac{1}{1 + e^{-(w \cdot c_i)}}$$

1. Введение
2. Счетные и нейронные модели представления слова
   - Факторизация матрицы терм-контекст
   - Word2Vec
   - Word2Vec как факторизация матрицы sPMI
3. Использование представлений слова
4. Другие модели
   - Word2Vec-f
   - Doc2Vec
   - GloVe
   - FastText
   - StarSpace
   - AdaGram
5. Другое
   - Двуязычные представления слов
   - HistWords
   - Составления предметных словарей эмоционально-окрашенных слов

Дистрибутивная семантика и Word2Vec [LG14b]

Результат Word2Vec: два матрицы, $E^W \in \mathbb{R}^{V_W \times d}$ и $E^C \in \mathbb{R}^{V_C \times d}$
Пусть $M' = E^W \times E^C$.
Связь исходной матрицы слово-контекст M и M':

$$w \cdot c = M'_{[w,c]} = \text{PMI}(w, c) - \log k,$$

где k – число отрицательных контекстов

Сравнение моделей эмбеддингов [SLMJ15]

1. Внутренние (intrinsic) задачи
   - Определение похожих слов
   - Определение аналогий
   - Категоризация слов
   - Определение лишнего слова
   - Определение объектов глаголов
2. Внешние (extrinsic) задачи
   - Классификация текстов
   - Извлечение именованных сущностей
   - Расширение запроса

Результаты зависят от использованного корпуса для обучения, гиперпараметров обучения, корпуса для тестирования. Невозможно определить модель эмбеддингов, превосходящую остальные.

Word2Vec-f (dependency embeddings) [LG14a]

Выбор контекста: синтаксически зависимые слова.

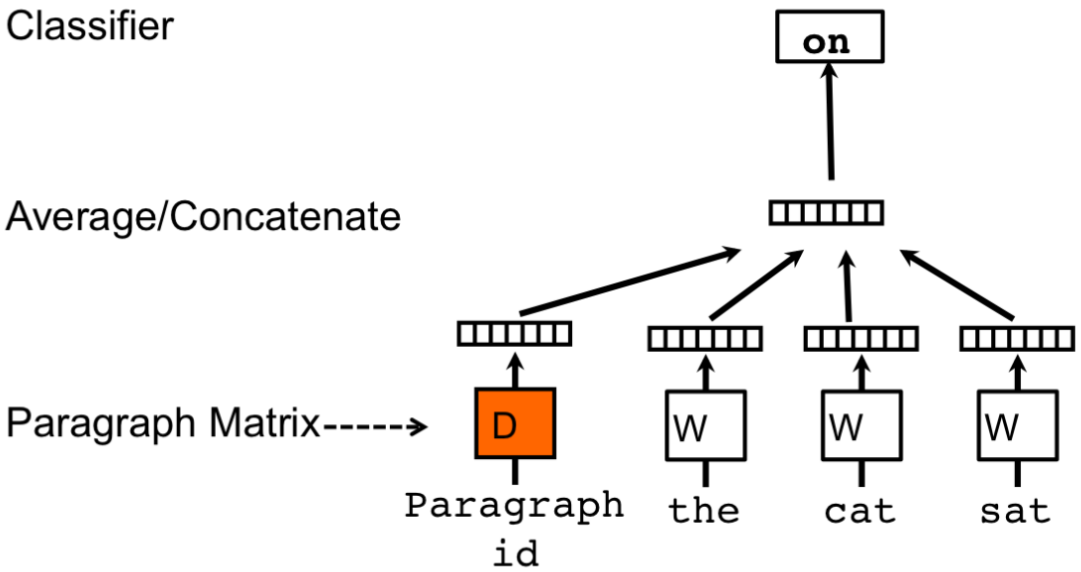
Результат: функциональные зависимости.

Target Word	BoW5	BoW2	Deps
batman	nightwing aquaman catwoman superman manhunter	superman superboy aquaman catwoman batgirl	superman superboy supergirl catwoman aquaman
hogwarts	dumbledore hallows half-blood malfoy snape	evernight sunnydale garderobe blandings collinwood	sunnydale collinwood calarts greendale millfield
turing	nondeterministic non-deterministic computability deterministic finite-state	non-deterministic finite-state nondeterministic buchi primality	pauling hotelling heting lessing hamming
florida	gainesville fla jacksonville tampa lauderdale	fla alabama gainesville tallahassee texas	texas louisiana georgia california carolina
object-oriented	aspect-oriented smalltalk event-driven prolog domain-specific	aspect-oriented event-driven objective-c dataflow 4gl	event-driven domain-specific rule-based data-driven human-centered
dancing	singing dance dances dancers tap-dancing	singing dance dances breakdancing clowning	singing rapping breakdancing miming busking

Насколько похожи два предложения (абзаца)? [LM14]

Как найти вектор-предложения (абзаца) ?

1. Усреднить вектора слов, входящих в каждое предложение (с $tf - idf$ весами)
2. Doc2vec: что word2vec, только для предложений (абзацев)



Global **Ve**ctors [PSM14]

	x = solid	x = gas	x = water	x = random
p(x|ice)	large	small	large	small
p(x|steam)	small	large	large	small
p(x|ice)/p(x|steam)	large	small	1	1

$$w \cdot c + b_{|w|} + b_{|c|} = \log(w, c) \ \forall (w, c) \in D,$$

$b_{|w|}, b_{|c|}$ – обучаемые сдвиги для слов и контекстов

FastText [BGJM16]

Слово w представляем символьными n-грамами:
$n = 3$, $G_{where} = \_wh, whe, her, re\_, \_where\_$
$sim_{w2v}(u, v) =< u, v >$
$sim_{ft}(u, v) = \sum_{e \in G_u} \sum_{g \in G_v} < e, v >$
git

StarSpace [WFC+17]

Находит вектора для различных сущностей:

$$\sum_{(a,b^+)\in E^+, b^-\in E^-} L^{batch}(sim(a,b^+), sim(a,b_1^-), \ldots, sim(a,b_k^-))$$

- $E+$ – генератор положительных (наблюдаемых) пар, зависит от задачи
- $E-$ – генератор отрицательных (ненаблюдаемых) пар
- sim – функция близости, косинусная или Евклидова
- L – функция потерь

Сценарии использования:

- Классификация текстов: a – документы, b – метки классов
- Поиск по запросу: a – запрос, b – документы

git

AdaGram [BKOV16]

Находит k смыслов слова. Обозначим смыслы через Z – все возможные смыслы всех слов, всего их N:

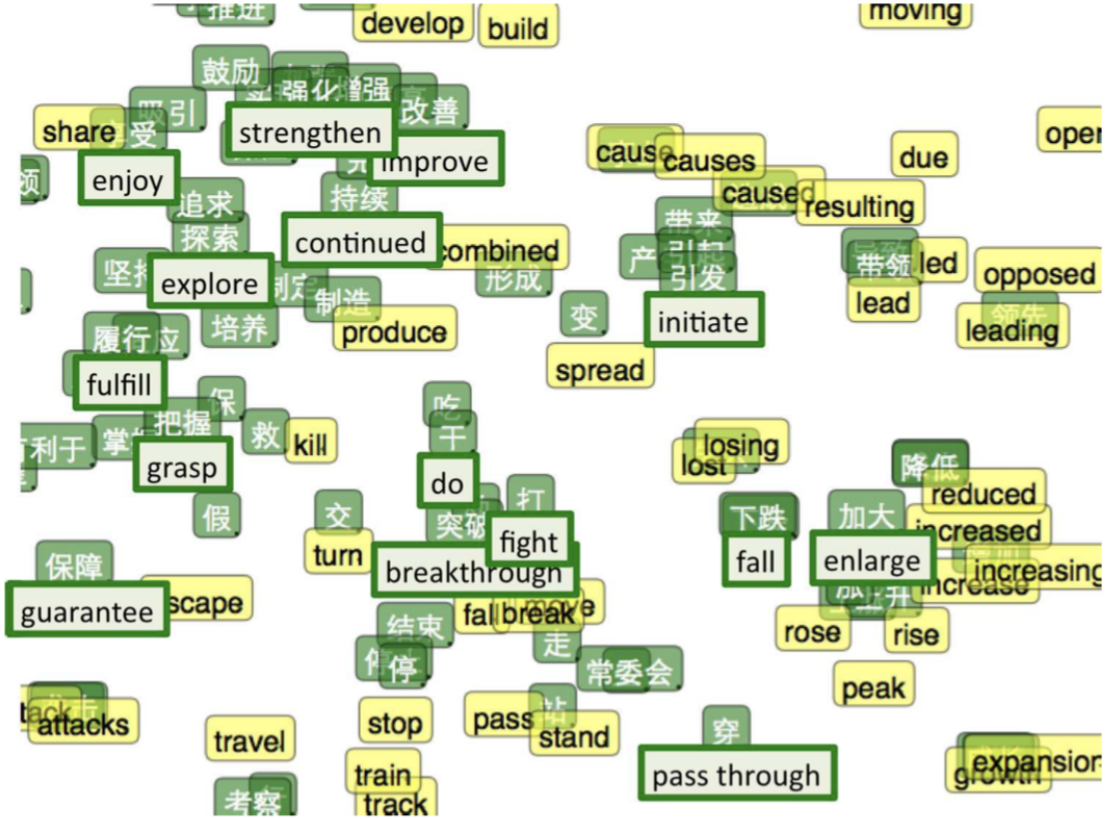
$$p(C, Z, \beta | X, \alpha, \theta) = \prod_{w \in V_w} \prod_{k}^{\infty} p(\beta_{wk} | \alpha) \prod_{i=1}^{N} [p(z_i | x_i, \beta) \prod_{c \in V_c} p(c | z_i, x_i, \theta)]$$

Демо

1. Введение
2. Счетные и нейронные модели представления слова
   - Факторизация матрицы терм-контекст
   - Word2Vec
   - Word2Vec как факторизация матрицы sPMI
3. Использование представлений слова
4. Другие модели
   - Word2Vec-f
   - Doc2Vec
   - GloVe
   - FastText
   - StarSpace
   - AdaGram
5. Другое
   - Двуязычные представления слов
   - HistWords
   - Составления предметных словарей эмоционально-окрашенных слов

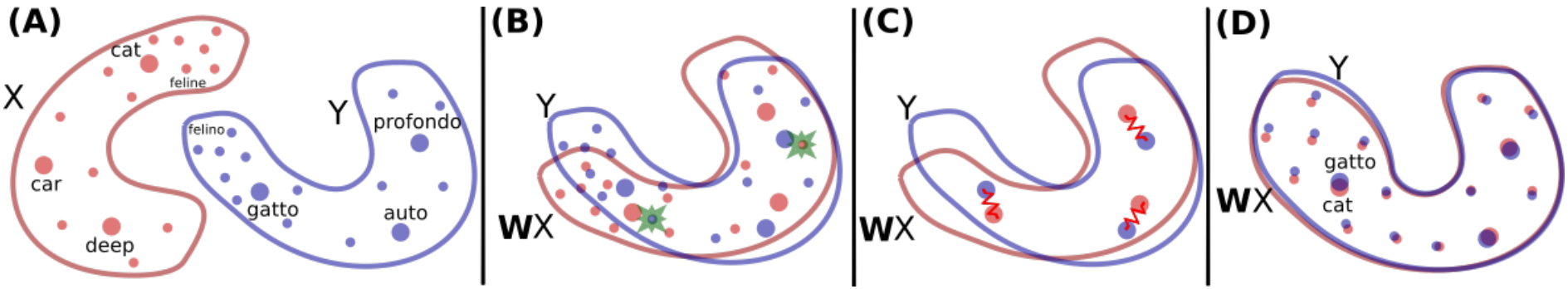
Двуязычные эмбедденги [ZSCM13]

Дан (выровненный) параллельный корпус. Контекст слова: перевод этого слова на другой язык.



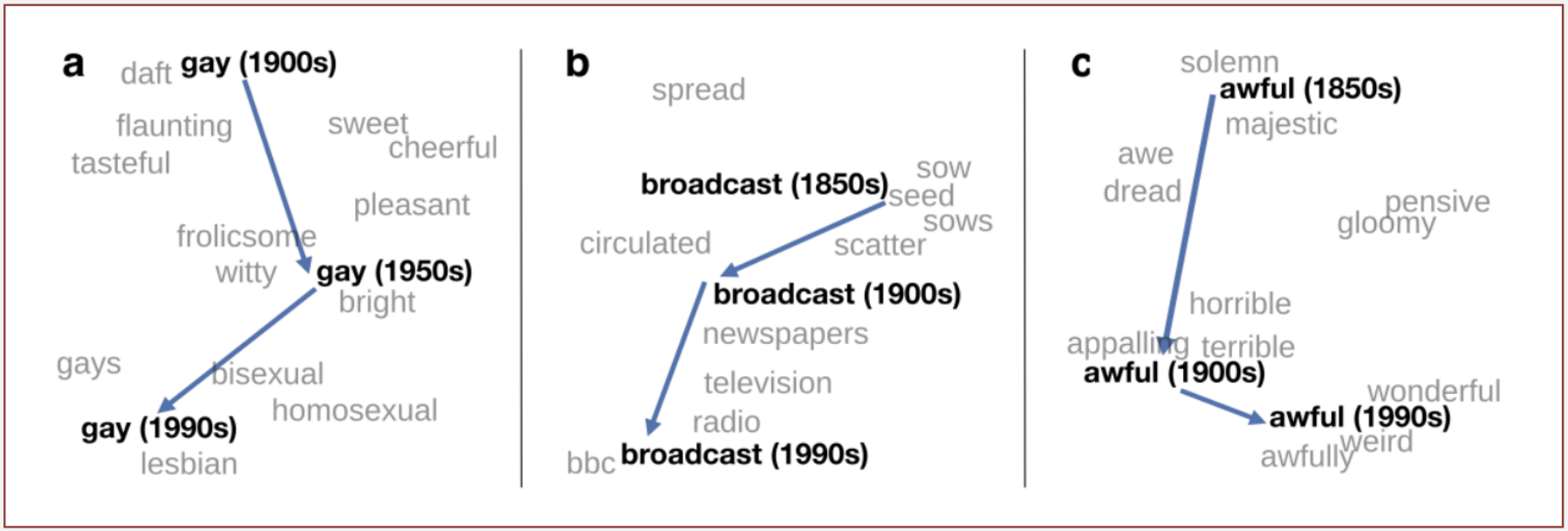
Двуязычные эмбедденги [CLR+17]

1. Дано два невыровненных пространства слов
2. Adversarial learning для определения матрицы поворота W
3. Прокрустово преобразование для уточнения W
4. $k - NN$-подобный метод для окончательного выравнивания



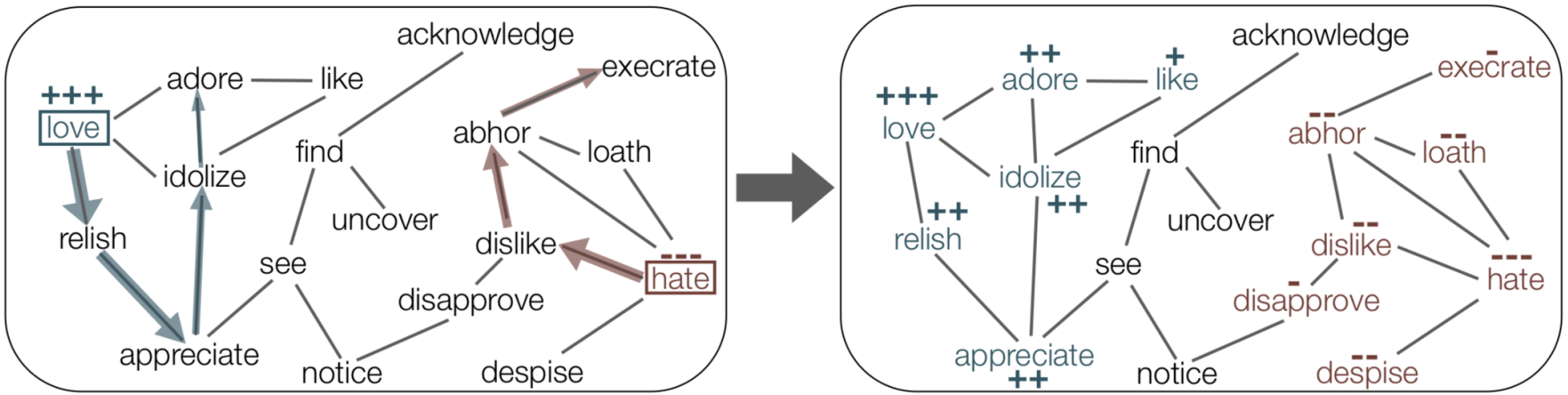
HistWords [HLJ16]

Диахронические эмбеддинги: Прокрустово преобразование для поворота пространства эмбеддингов из периода $t-1$ в t



Составления предметных словарей эмоционально-окрашенных слов [HCLJ16]

1. Граф близости на словах
2. Случайное блуждание для распространения метки



a. Run random walks from seed words.

b. Assign polarity scores based on frequency of random walk visits.

Источники I


Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov, *Enriching word vectors with subword information*, arXiv preprint arXiv:1607.04606 (2016).


Sergey Bartunov, Dmitry Kondrashkin, Anton Osokin, and Dmitry Vetrov, *Breaking sticks and ambiguities with adaptive skip-gram*, Artificial Intelligence and Statistics, 2016, pp. 130–138.


Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou, *Word translation without parallel data*, arXiv preprint arXiv:1710.04087 (2017).


Ronan Collobert and Jason Weston, *A unified architecture for natural language processing: Deep neural networks with multitask learning*, Proceedings of the 25th international conference on Machine learning, ACM, 2008, pp. 160–167.

Источники II


William L Hamilton, Kevin Clark, Jure Leskovec, and Dan Jurafsky, *Inducing domain-specific sentiment lexicons from unlabeled corpora*, Proceedings of the Conference on Empirical Methods in Natural Language Processing. Conference on Empirical Methods in Natural Language Processing, vol. 2016, NIH Public Access, 2016, p. 595.


William L Hamilton, Jure Leskovec, and Dan Jurafsky, *Diachronic word embeddings reveal statistical laws of semantic change*, arXiv preprint arXiv:1605.09096 (2016).


Andrey Kutuzov and Elizaveta Kuzmenko, *Webvectors: a toolkit for building web interfaces for vector semantic models*, International Conference on Analysis of Images, Social Networks and Texts, Springer, 2016, pp. 155–161.

Источники III


Omer Levy and Yoav Goldberg, *Dependency-based word embeddings*, Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), vol. 2, 2014, pp. 302–308.


_____, *Neural word embedding as implicit matrix factorization*, Advances in neural information processing systems, 2014, pp. 2177–2185.


Quoc Le and Tomas Mikolov, *Distributed representations of sentences and documents*, International Conference on Machine Learning, 2014, pp. 1188–1196.


Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean, *Efficient estimation of word representations in vector space*, arXiv preprint arXiv:1301.3781 (2013).

Источники IV


Jeffrey Pennington, Richard Socher, and Christopher Manning, *Glove: Global vectors for word representation*, Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 2014, pp. 1532–1543.


Tobias Schnabel, Igor Labutov, David Mimno, and Thorsten Joachims, *Evaluation methods for unsupervised word embeddings*, Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 2015, pp. 298–307.


Joseph Turian, Lev Ratinov, and Yoshua Bengio, *Word representations: a simple and general method for semi-supervised learning*, Proceedings of the 48th annual meeting of the association for computational linguistics, Association for Computational Linguistics, 2010, pp. 384–394.

Источники V


L. Wu, A. Fisch, S. Chopra, K. Adams, A. Bordes, and J. Weston, *Starspace: Embed all the things!*, arXiv preprint arXiv:1709.03856 (2017).


Will Y Zou, Richard Socher, Daniel Cer, and Christopher D Manning, *Bilingual word embeddings for phrase-based machine translation*, Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, 2013, pp. 1393–1398.