курс «Глубокое обучение»


# Векторные представления слов и текстов

Александр Дьяконов

4 апреля 2022 года

# План

**классические способы представления слов**

OHE, counts, LSA, кластеризация, LDA

**DL-классика**

word2vec, fasttext, Glove

**учёт контекста**

CoVe, ELMo, FLAIR

**представление текстов**

Doc2Vec / paragraph2vec, The skip-thoughts model,
Autoencoder pretraining, StarSpace, DAN
Universal Sentence Encoder

**DSSM**

# Представления слов

**решают проблему «что дать на вход сети»**

токен / номер $\rightarrow$ вектор

способ «засунуть» дискретные объекты в НС

**представляют слова так, что похожие слова имеют похожие представления**

решается проблема незнакомых слов

**имеют небольшую размерность**

$\Rightarrow$ сокращают число параметров сети

**могут быть получены на большом неразмеченном тексте**

трансферное обучение

# Способы кодирования / представления слов

- **OHE**

слишком большая размерность, нет хорошей близости

- **counts (сумма OHE соседей)**

более нетривиальная оценка близости с помощью cos

- **вложение (embeddings)**

умный алгоритм задания кодировки

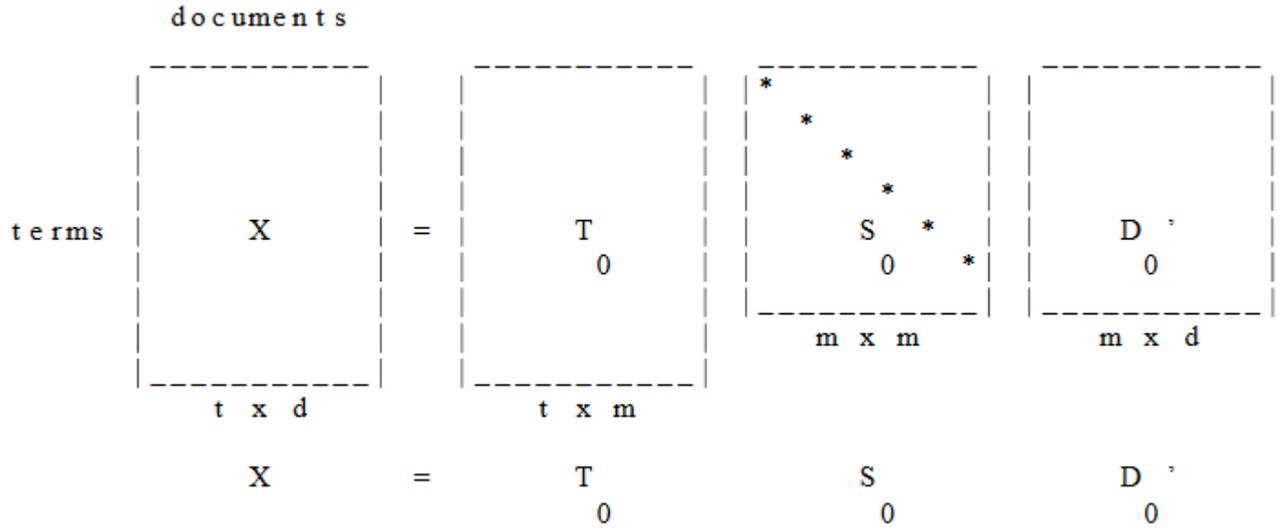
## «word embeddings»

**Представления слов в вещественном многомерном пространстве**

**$\Rightarrow$ можно использовать в матмоделях**

**Предобученные** **Обученные для конкретной задачи**

# Классические способы представления слов: LSA




S. Deerwester «Indexing by latent semantic analysis», 1990

http://lsa.colorado.edu/papers/JASIS.lsi.90.pdf

# Классические способы представления слов: кластеризация слов

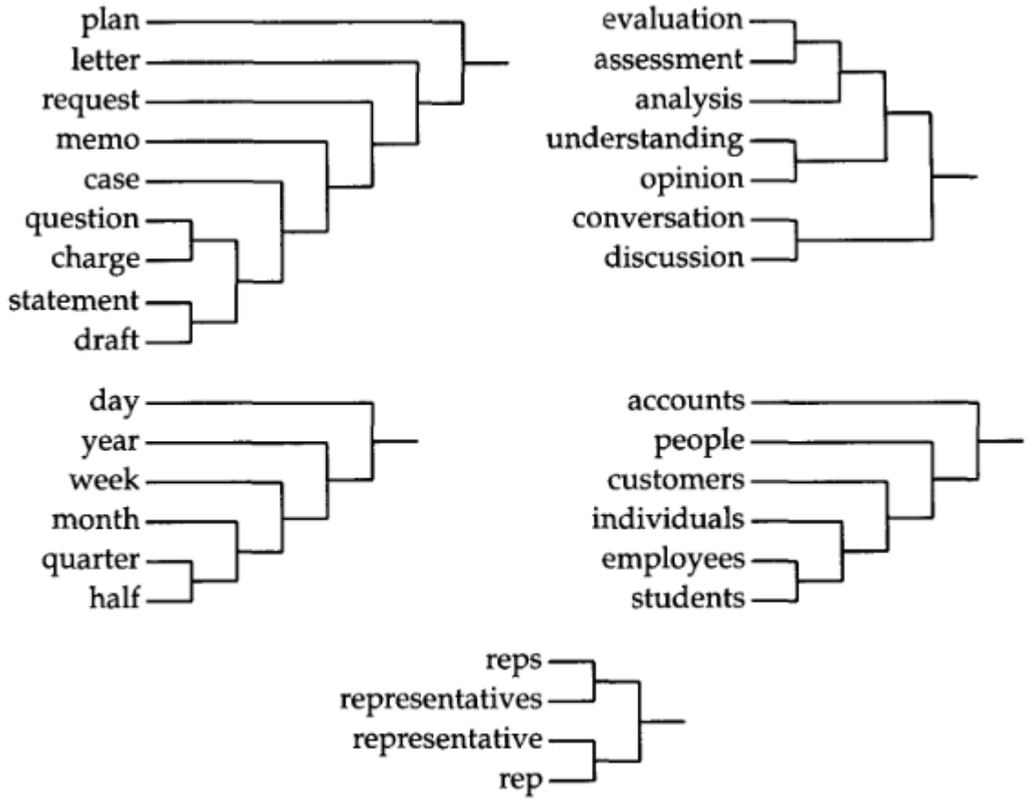



**Figure 2**
Sample subtrees from a 1,000-word mutual information tree.

**Peter F. Brown et. al. «Class-Based n-gram Models of Natural Language»**
**https://www.aclweb.org/anthology/J92-4003.pdf**

# Классические способы представления слов: LDA

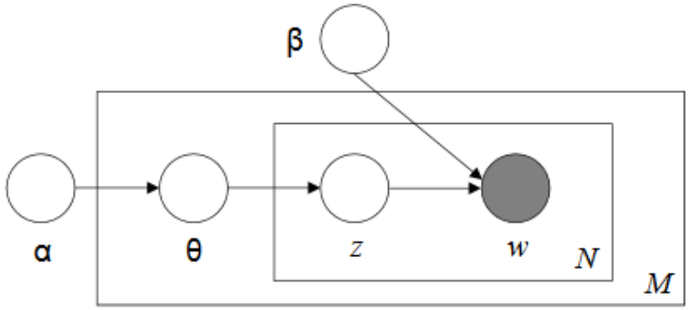



Figure 1: Graphical model representation of LDA. The boxes are "plates" representing replicates. The outer plate represents documents, while the inner plate represents the repeated choice of topics and words within a document.

| "Arts" | "Budgets" | "Children" | "Education" |
|---|---|---|---|
| NEW | MILLION | CHILDREN | SCHOOL |
| FILM | TAX | WOMEN | STUDENTS |
| SHOW | PROGRAM | PEOPLE | SCHOOLS |
| MUSIC | BUDGET | CHILD | EDUCATION |
| MOVIE | BILLION | YEARS | TEACHERS |
| PLAY | FEDERAL | FAMILIES | HIGH |
| MUSICAL | YEAR | WORK | PUBLIC |
| BEST | SPENDING | PARENTS | TEACHER |
| ACTOR | NEW | SAYS | BENNETT |
| FIRST | STATE | FAMILY | MANIGAT |
| YORK | PLAN | WELFARE | NAMPHY |
| OPERA | MONEY | MEN | STATE |
| THEATER | PROGRAMS | PERCENT | PRESIDENT |
| ACTRESS | GOVERNMENT | CARE | ELEMENTARY |
| LOVE | CONGRESS | LIFE | HAITI |

**D.M. Blei «Latent Dirichlet Allocation» // Journal of Machine Learning, 2003**

http://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf

The William Randolph Hearst Foundation will give $1.25 million to Lincoln Center, Metropolitan Opera Co., New York Philharmonic and Juilliard School. "Our board felt that we had a real opportunity to make a mark on the future of the performing arts with these grants an act every bit as important as our traditional areas of support in health, medical research, education and the social services," Hearst Foundation President Randolph A. Hearst said Monday in announcing the grants. Lincoln Center's share will be $200,000 for its new building, which will house young artists and provide new public facilities. The Metropolitan Opera Co. and New York Philharmonic will receive $400,000 each. The Juilliard School, where music and the performing arts are taught, will get $250,000. The Hearst Foundation, a leading supporter of the Lincoln Center Consolidated Corporate Fund, will make its usual annual $100,000 donation, too.

# Для чего использовались: n-граммная языковая модель

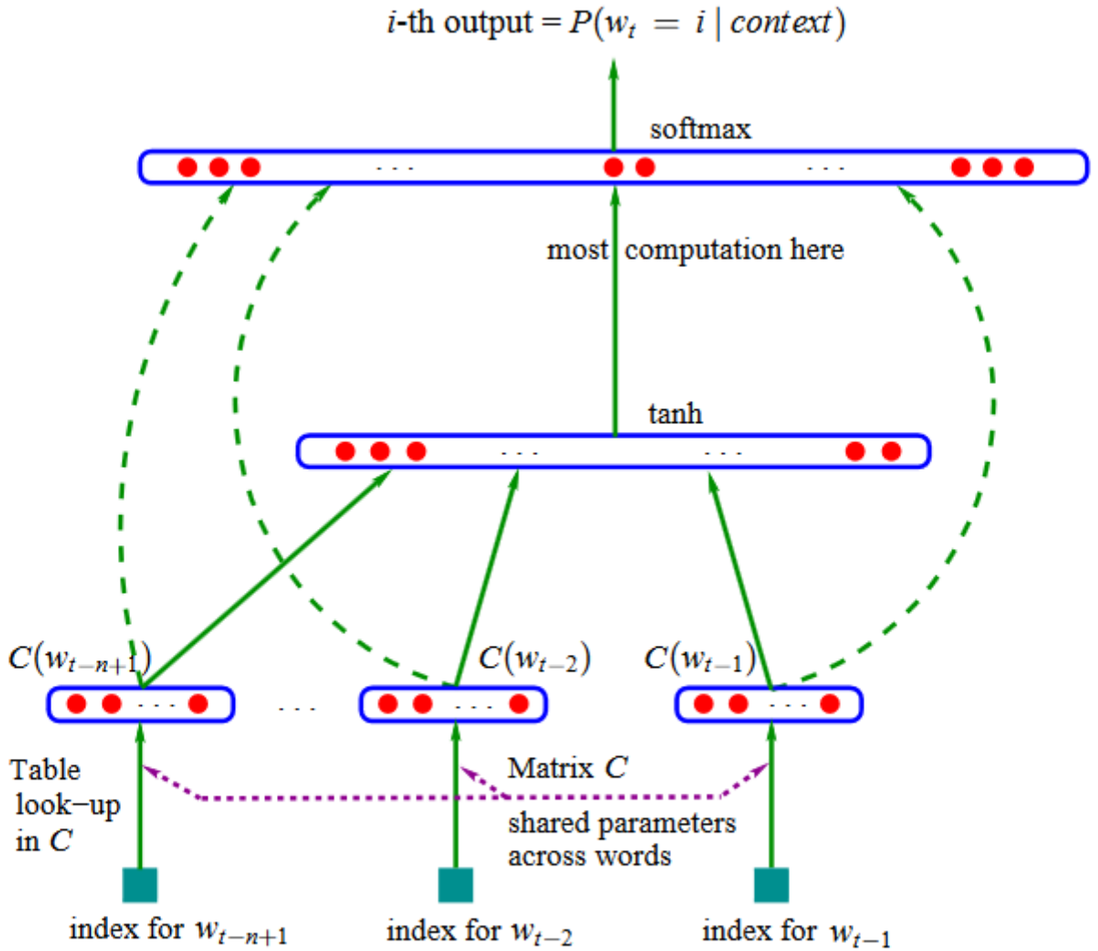
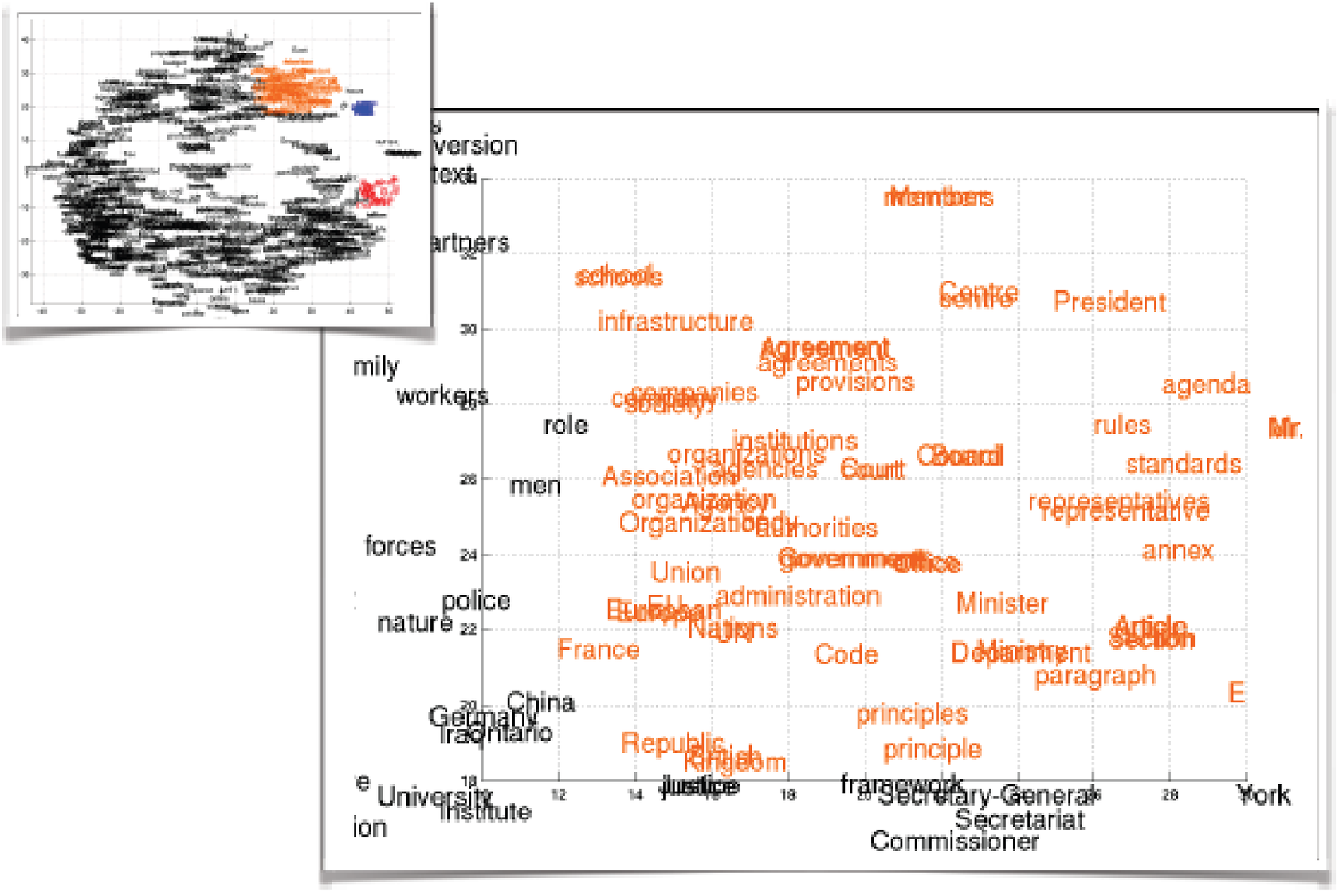



Figure 1: Neural architecture: $f(i, w_{t-1}, \cdots, w_{t-n+1}) = g(i, C(w_{t-1}), \cdots, C(w_{t-n+1}))$ where $g$ is the neural network and $C(i)$ is the $i$-th word feature vector.

**Yoshua Bengio «Neural Probabilistic Language Model» Journal of Machine Learning Research**

**http://www.jmlr.org/papers/volume3/bengio03a/bengio03a.pdf**

# Вложение слов в непрерывное пространство (embedding)



# DL-классика: безконтекстные методы

**context-free – не учитывающие контекст**

**учитывают контекст при обучении представления,**
**но при использовании это уже фиксированный вектор –**
**контекст не учитывается**

- **word2vec** = предсказания слово $\leftrightarrow$ контекст
- **fasttext** = word2vec + ngrams
- **Glove** = разложение матрицы совместной встречаемости

# word2vec – дистрибутивная семантика

**Трюк: настраиваем модель, но не для использования в задаче,**
**которой учим (нас интересуют формируемые внутренние представления)**
**принцип трансферного обучения (ex: автокодировщики)**

**Термины «distributional semantics»**

**Смысл слова определяется контекстом**

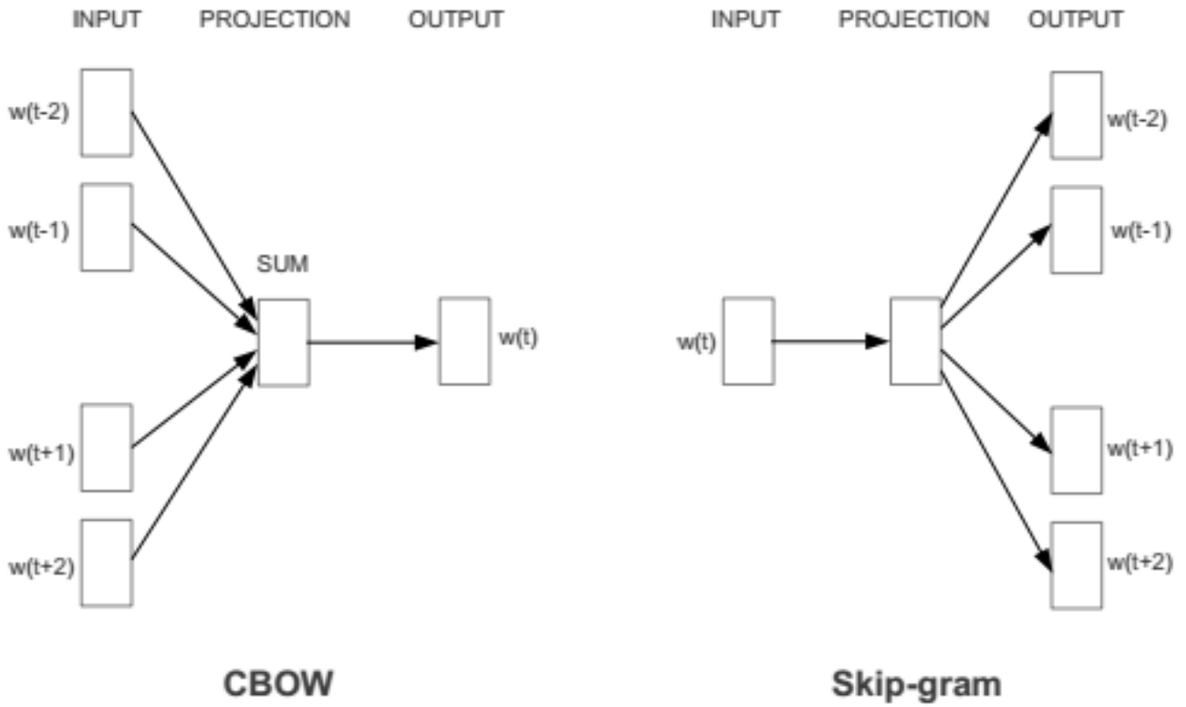
**Полосатая маленькая ***** мурлычит и пьёт молоко**

**Весна**

**Ручьи**
**Тает**
**Цветёт**
**Зеленеет**
**Прилетают**

**[Mikolov et al. 2013]**

# word2vec: два подхода к реализации




**CBOW = Continuous Bag of Words (быстрее, окно ~ 5, большие корпуса)**
**skipgram model (раньше считалось, что лучше, окно ~ 10, небольшие корпуса)**

**это пример самообучения (self-supervision) – когда разметка автоматическая**
**можно использовать большие корпуса внешних текстов**

# word2vec: CBOW

**Предсказываем слово по контексту**

**используется реже, чем следующая реализация**

$$P(x_t \mid \text{context}(x_t)) = \text{softmax}\left( V \left( W \sum_{x_i \in \text{context}(x_t)} OHE(x_i) \right) \right)$$

**выделено то, что будем считать кодировкой**

**контекст – слово (слова), которое недалеко располагается (в окрестности)**

http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/

# word2vec: CBOW

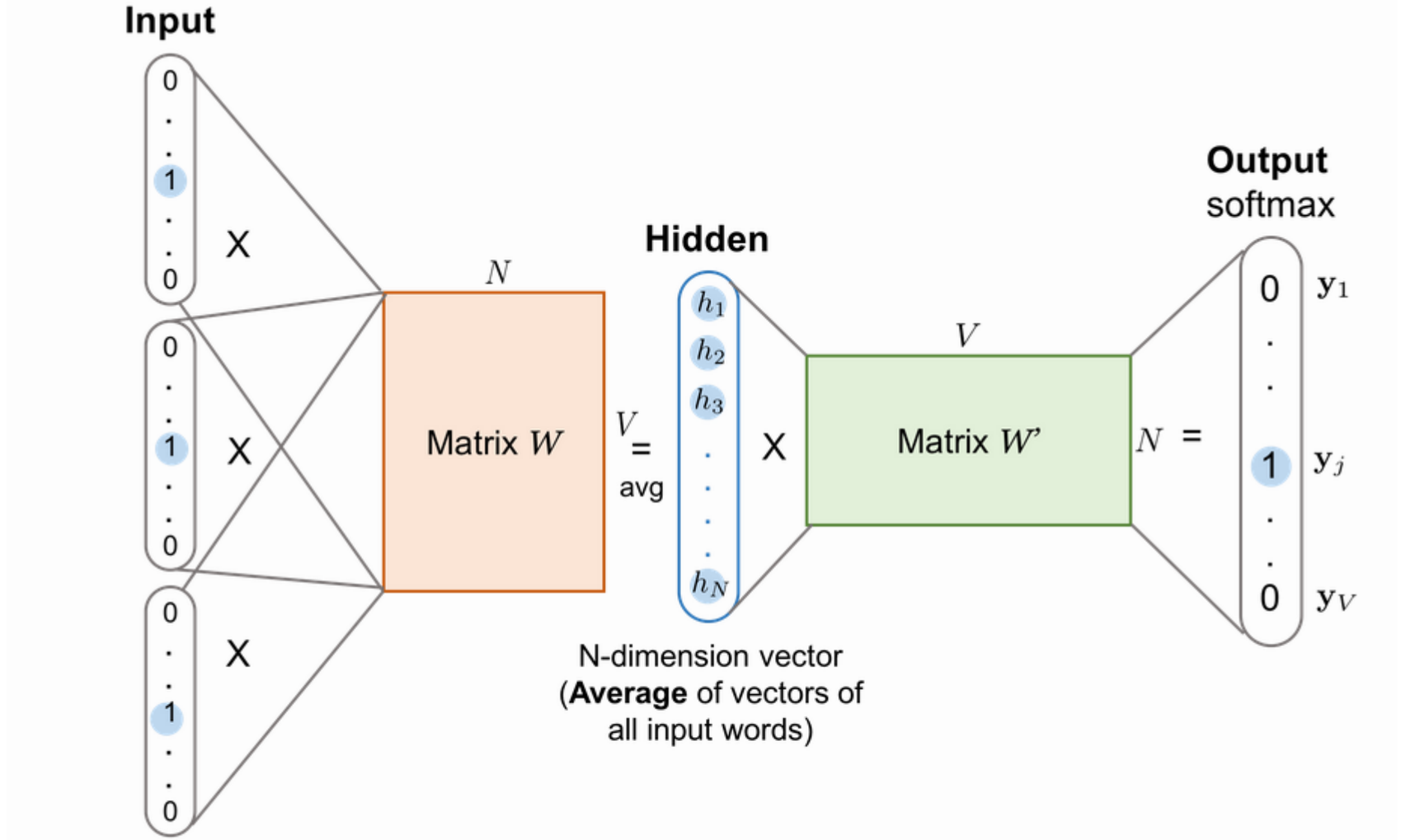
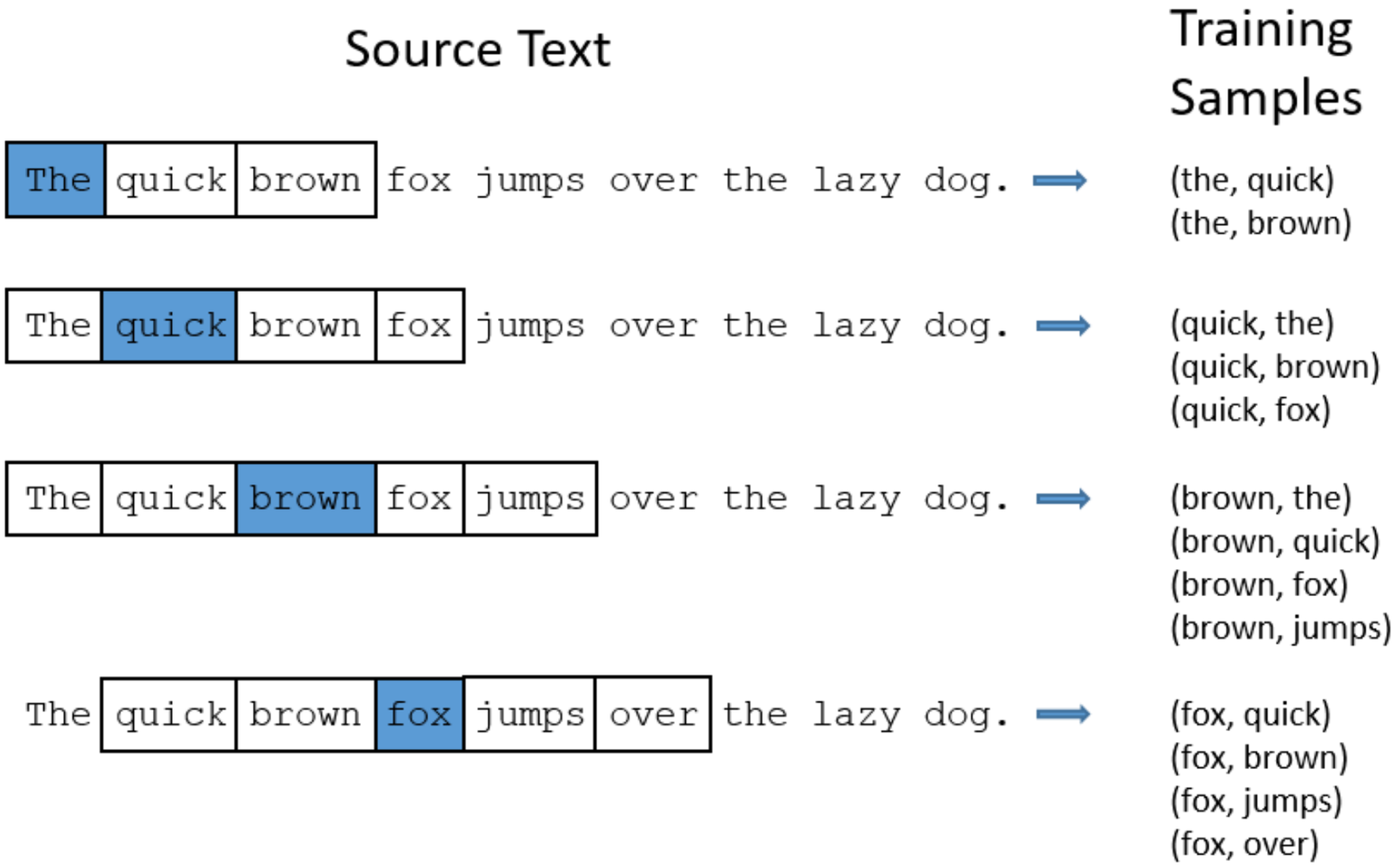



*Fig. 2. The CBOW model. Word vectors of multiple context words are averaged to get a fixed-length vector as in the hidden layer. Other symbols have the same meanings as in Fig 1.*

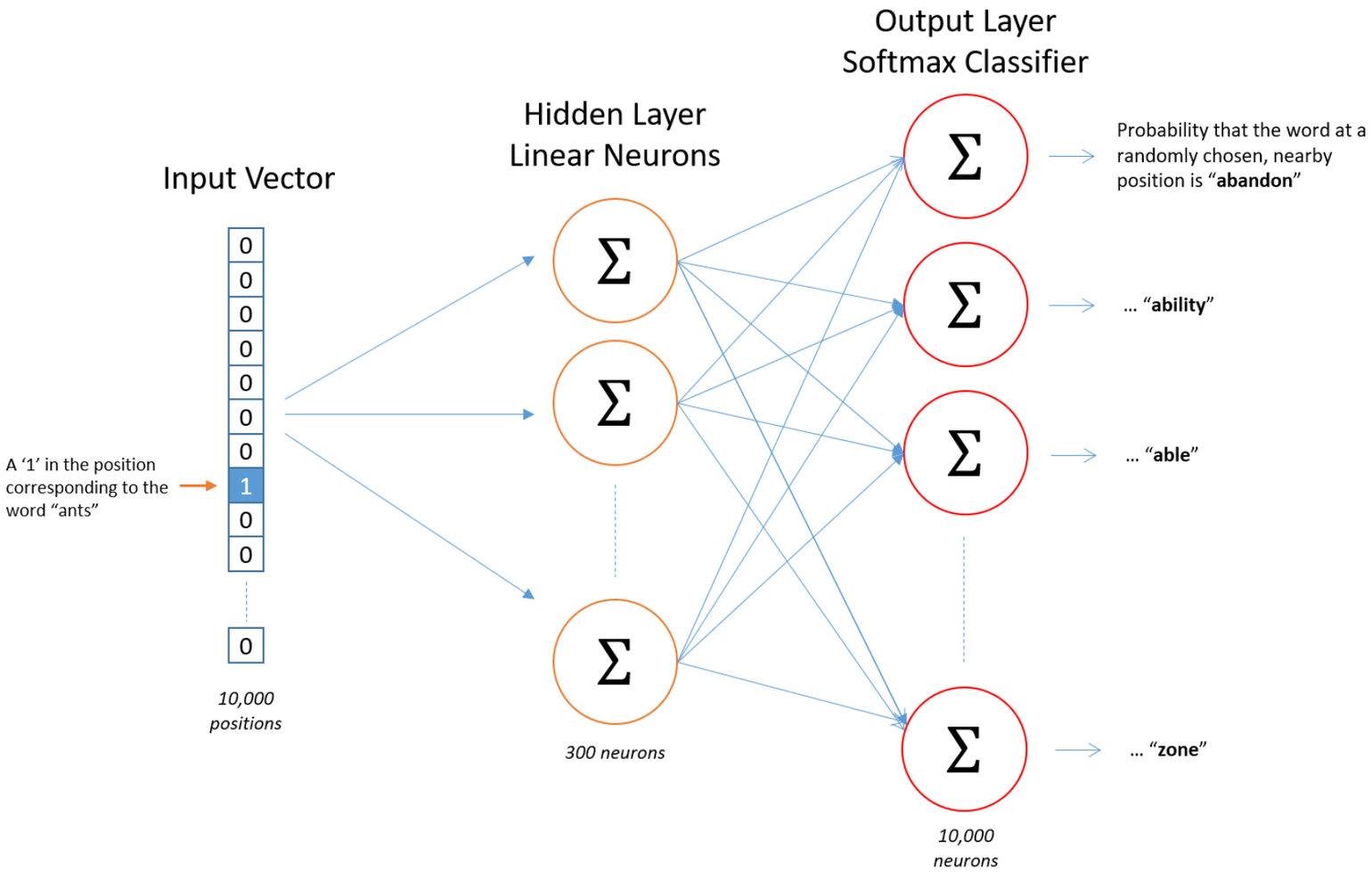
https://lilianweng.github.io/lil-log/2017/10/15/learning-word-embedding.html

# word2vec: skip-gram

## Предсказываем контекст по слову: слово → слово



# word2vec: skip-gram




**вход:** ONE-кодировка слова

**выход:** распределение вероятностей

**средний слой** – для нашего кодирования

# word2vec: skip-gram

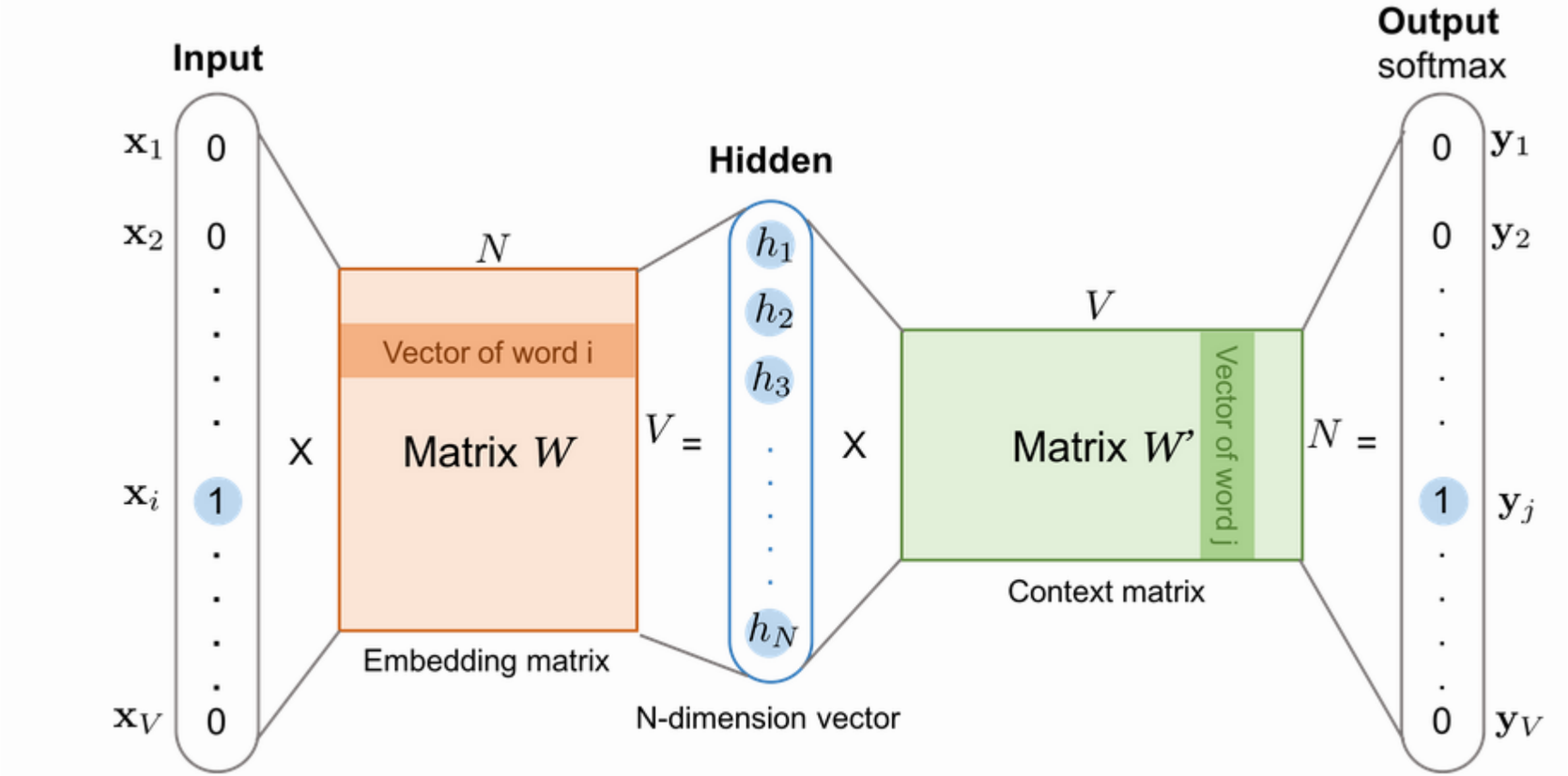



*Fig. 1. The skip-gram model. Both the input vector* $\mathbf{x}$ *and the output* $\mathbf{y}$ *are one-hot encoded word representations. The hidden layer is the word embedding of size* $N$.

https://lilianweng.github.io/lil-log/2017/10/15/learning-word-embedding.html

# word2vec

**Огромная НС**

**Первый слой – #слов × размерность представления (~300)**

**Как обучать????**

здесь предложены модификации обучения:

Mikolov T. «Distributed Representations of Words and Phrases and their Compositionality» //

https://arxiv.org/pdf/1310.4546.pdf

**не только уменьшают время обучения, но и улучшают качество представлений**

**/ код слова = строка первой матрицы + столбец второй**

**или строка первой матрицы**

Следующие слайды по

http://mccormickml.com/2017/01/11/word2vec-tutorial-part-2-negative-sampling/

**Есть отличия между реализацией и статьёй!**

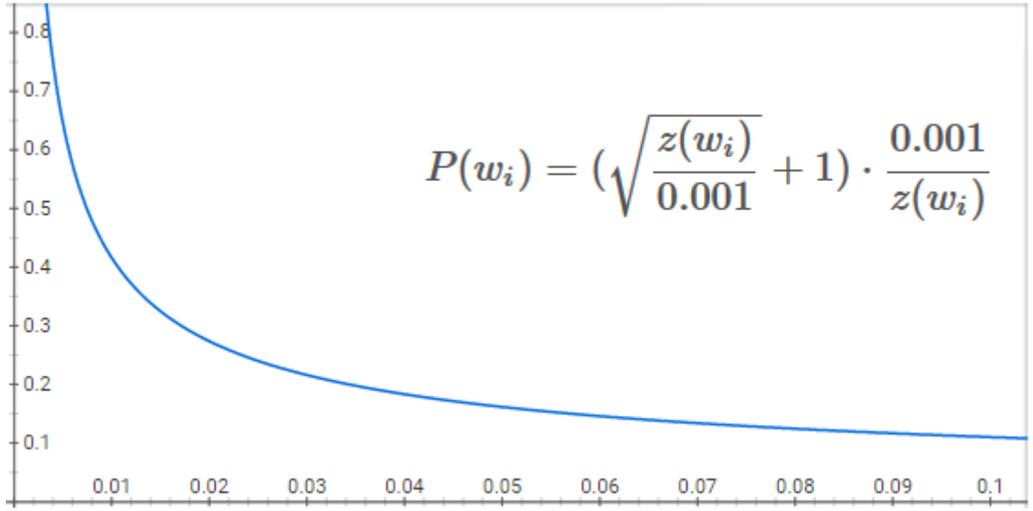
# word2vec

**Распространённые фразы – одно слово**

White_Spunner_Construction
Bad_Habits
Toxics_Alliance

**Частые слова – реже выбираются при обучении**

вероятность быть выбранным от частоты:




**(quick, the)**
**– про эти слова речь**
используются не все пары
– идёт сэмплирование

**«Negative Sampling»**

у («открыл») = OHE(«дверь»)
чтобы не править много выходов, соответствующим нулям,
выбираем несколько случайных (5–20)

# word2vec – немного математики

**Последовательность слов** $x_1, \ldots, x_T$

**Правдоподобие**

$$\prod_{t=1}^{T} \prod_{c \in C_t} p(x_c \mid x_t) \sim \sum_{t=1}^{T} \sum_{c \in C_t} \log p(x_c \mid x_t) \to \max$$

**(второе произведение по окрестности – индексы соседних слов)**

**Можно:** $p(x_c \mid x_t) = \dfrac{\exp(s(x_t, x_c))}{\sum_x \exp(s(x_t, x))}$

**Такая модель подходила бы,**
**если бы для каждого слова один правильный ответ**
**хотя тоже используется**

# word2vec: Negative Sampling

**Как делаем… «skipgram model with negative sampling»** [Mikolov]

**Используем «negative log-likehood»**

$$\log\left(1+\exp(-s(x_t,x_c))\right)+\sum_{x\in N_{t,c}}\log\left(1+\exp(s(x_t,x))\right)$$

$N_{t,c}$ **– выборка негативных примеров**

**Если logloss** $l(z)=\log\left(1+\exp(-z)\right)$**, то**

$$\sum_{t=1}^{T}\left[\sum_{c\in C_t}l(s(x_t,x_c))+\sum_{x\in N_{t,c}}l(-s(x_t,x))\right]\to\min$$

**Скоринговая функция:** $s(x_t,x_c)=\text{vec}(x_t)^{\text{T}}\cdot\text{vec}(x_c)$

**– тут представление входа и выхода**

**тут нужны будут негативные примеры**

# word2vec: Negative Sampling

**Выбор негативных слов производится не равномерно**

**тогда вероятность выбора слова**

$$\frac{\#i}{\sum_j \#j}$$

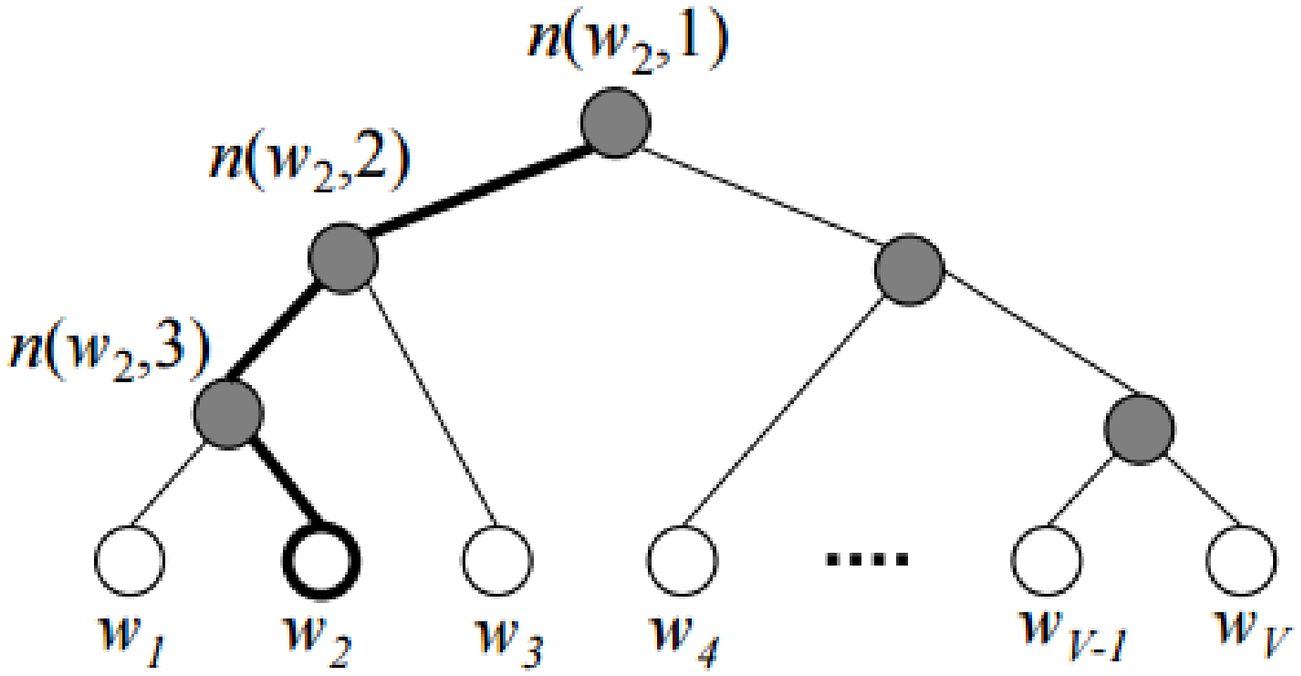
**а с вероятностью**

$$\frac{(\#i)^{3/4}}{\sum_j (\#j)^{3/4}}$$

**в результате экспериментов – так лучше**

# Hierarchical Softmax

**softmax-слой представляется так (специальная кодировка Хаффмана)**




**листья – слова**

**вероятность = произведение вероятностей в рёбрах пути**

# Ближайшие соседи

| | | |
|---|---|---|
| Peace | Path | Stop |
| Peaceful | Paths | Quit |
| Friendship | Approach | Stopped |
| Nonviolence | Titled | Avoid |
| | Pathway | Resist |
| | Way | |

http://bionlp-www.utu.fi/wv_demo/

**+ осмысленное соседство**

**+ осмысленные арифметические операции**

# Операции над представлениями слов

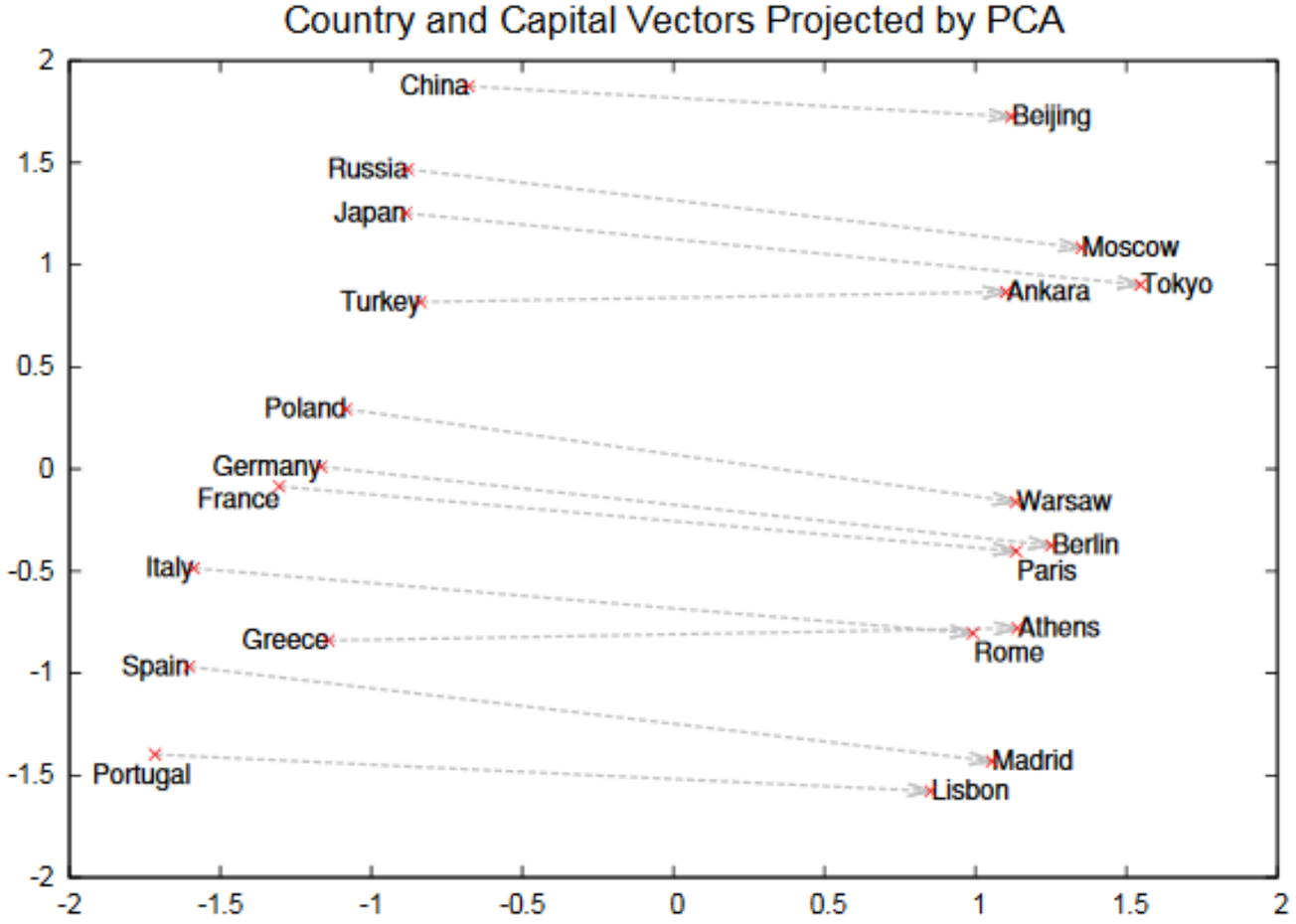



Figure 2: Two-dimensional PCA projection of the 1000-dimensional Skip-gram vectors of countries and their capital cities. The figure illustrates ability of the model to automatically organize concepts and learn implicitly the relationships between them, as during the training we did not provide any supervised information about what a capital city means.

[Mikolov et al., 2013] https://arxiv.org/pdf/1310.4546.pdf

# word2vec: ошибка в исходном коде

**Считалось, что CBOW хуже Skip-gram**

**Оказалось, что была ошибка в negative sampling в популярных реализациях**


$$\frac{\partial \mathcal{L}}{\partial v_{w_j}} = \boxed{\frac{1}{C}} \left[ (\sigma({v'_{w_O}}^{\top} v_c) - 1) v'_{w_O} + \sum_{i=1}^{k} \sigma({v'_{n_i}}^{\top} v_c) v'_{n_i} \right]$$

Figure 2: Average negative sampling loss per token for every batch of 5 million tokens for a single epoch of CBOW training on Wikipedia. The shaded region corresponds to the 95% bootstrapped confidence interval over average token loss on 100K token batches.

- **а в реализациях длина контекста C тоже сэмплируется!**
- **и ещё в производной по другому параметру C нет,**

**так что у нас получается смещённый вектор**

Ozan İrsoy et al. «Corrected CBOW Performs as well as Skip-gram» // https://arxiv.org/pdf/2012.15332.pdf

# Другие представления: fasttext

**тоже «слово → контекст»**

**попытка учесть морфологию слов**

**раньше «сеть», «сетевой», «сетью» разные векторы…**

**+ использовать n-граммные представления слова**

**«where» ~ <wh, whe, her, ere, re>**

**n-граммы хэшируются;)**

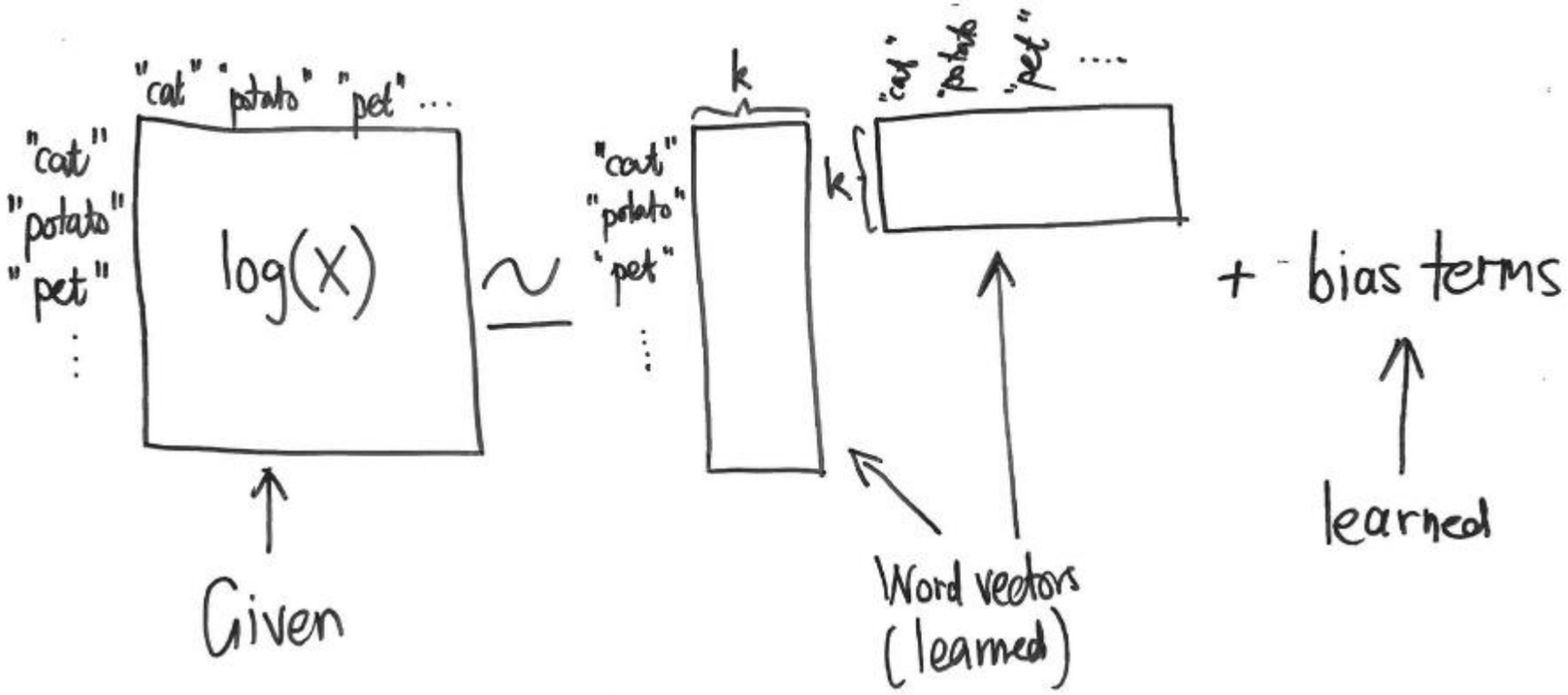
**код = сумма кодов для n-грамм**

**решается проблема новых слов**

Bojanowski P. et al. «Enriching Word Vectors with Subword Information» //
https://arxiv.org/pdf/1607.04606.pdf

https://fasttext.cc – тут есть все ссылки!!!

# Glove: Global Vectors for Word Representation




**идея в разложении матрицы**

http://building-babylon.net/2015/07/29/glove-global-vectors-for-word-representations/

https://nlp.stanford.edu/projects/glove/

# Glove: Global Vectors for Word Representation

$\#ij$ – **сколько раз слово j в контексте слова i**

**(на расстоянии ≤k слов)** **есть и другие варианты**

$$\sum_{i,j} f(\#ij)(w_i^{\mathrm{T}}\tilde{w}_j + b_i + \tilde{b}_j - \log(\#ij))^2 \to \min$$

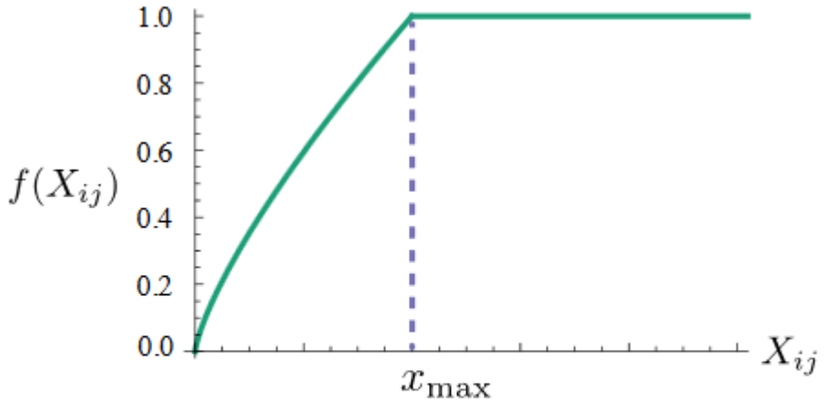
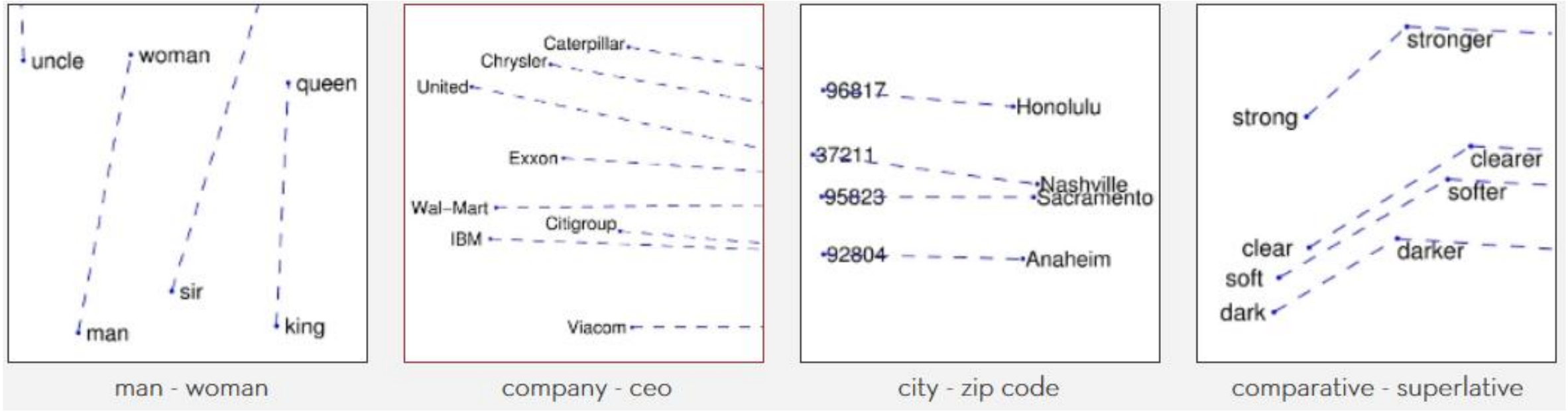



Figure 1: Weighting function $f$ with $\alpha = 3/4$.

$$f(x) = \begin{cases} \left(\dfrac{x}{x_{\max}}\right)^{\alpha}, & x < x_{\max}, \\ 1, & x \geq x_{\max}. \end{cases}$$

# Glove: ближайшие соседи

**frog**
**frogs**
**toad**
**litoria**
**leptodactylidae**
**rana**
**lizard**
**leutherodactylus**

# Проверка представлений

**1) визуально смотрим проекцию в пространстве, кластеры слов и т.п.**

**2) ближайшие соседи**

**есть бенчмарки схожих слов**

https://nlp.stanford.edu/~lmthang/data/papers/conll13_morpho.pdf

**задача «найди лишнее слово»**

**3) проверка арифметики**

**«король – мужчина + женщина = королева»**

**есть бенчмарки аналогий** https://arxiv.org/pdf/1301.3781.pdf

**кстати, в разных языках одинаковые линейные отношения**

**это позволяет наложить представление одного языка на другое!**

**4) качество при решении задач ML (downstream tasks)**

**ex: классификация тональности текстов фиксированным классификатором**

# Проверка представлений

## Словарь аналогий

Table 1: *Examples of five types of semantic and nine types of syntactic questions in the Semantic-Syntactic Word Relationship test set.*

| Type of relationship | Word Pair 1 | | Word Pair 2 | |
|---|---|---|---|---|
| Common capital city | Athens | Greece | Oslo | Norway |
| All capital cities | Astana | Kazakhstan | Harare | Zimbabwe |
| Currency | Angola | kwanza | Iran | rial |
| City-in-state | Chicago | Illinois | Stockton | California |
| Man-Woman | brother | sister | grandson | granddaughter |
| Adjective to adverb | apparent | apparently | rapid | rapidly |
| Opposite | possibly | impossibly | ethical | unethical |
| Comparative | great | greater | tough | tougher |
| Superlative | easy | easiest | lucky | luckiest |
| Present Participle | think | thinking | read | reading |
| Nationality adjective | Switzerland | Swiss | Cambodia | Cambodian |
| Past tense | walking | walked | swimming | swam |
| Plural nouns | mouse | mice | dollar | dollars |
| Plural verbs | work | works | speak | speaks |

# Семантический сдвиг

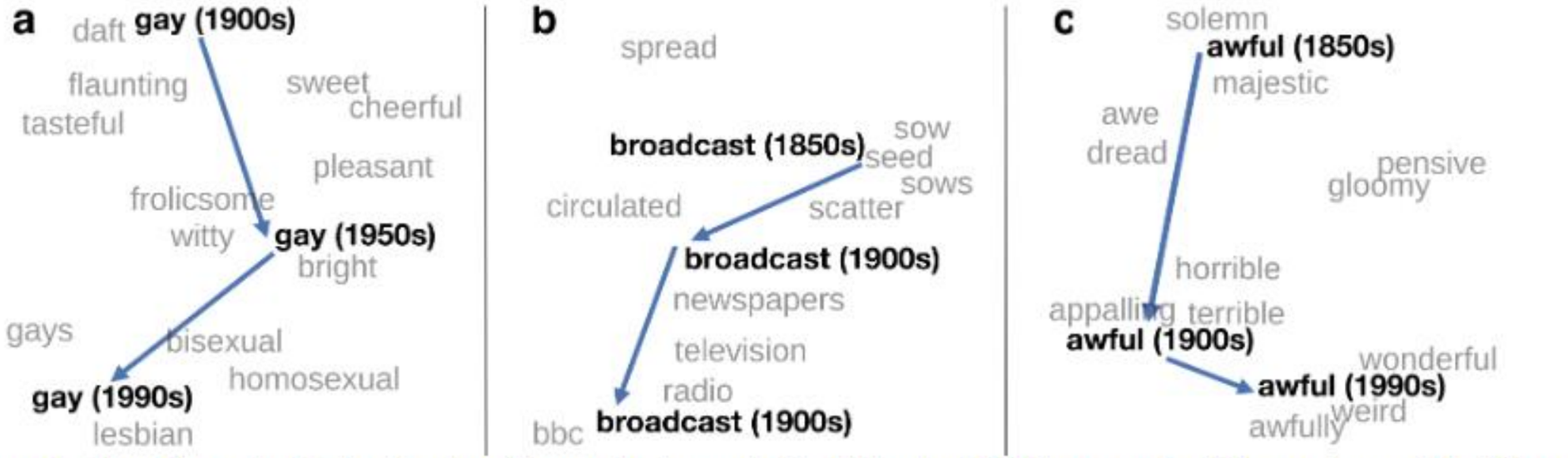



**Figure 1:** Two-dimensional visualization of semantic change in English using SGNS vectors.[2] **a**, The word *gay* shifted from meaning "cheerful" or "frolicsome" to referring to homosexuality. **b**, In the early 20th century *broadcast* referred to "casting out seeds"; with the rise of television and radio its meaning shifted to "transmitting signals". **c**, *Awful* underwent a process of pejoration, as it shifted from meaning "full of awe" to meaning "terrible or appalling" (Simpson et al., 1989).

**обучаем модель на текстах разных периодов**

**«выравниваем» пространства линейным/ортогональным преобразованием**

**логично применять растяжение и поворот**

William L. Hamilton, et al. «Diachronic Word Embeddings Reveal Statistical Laws of Semantic Change» // https://www.aclweb.org/anthology/P16-1141.pdf

# «Наложение языков»

## Ingredients:

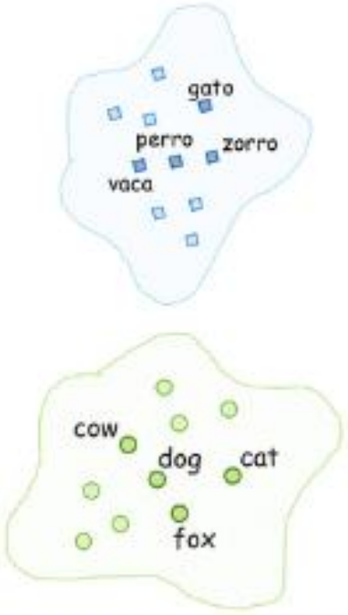
- corpus in one language (e.g., English)
- corpus in another language (e.g., Spanish)

- very small dictionary

cat ↔ gato
cow ↔ vaca
dog ↔ perro
fox ↔ zorro
...

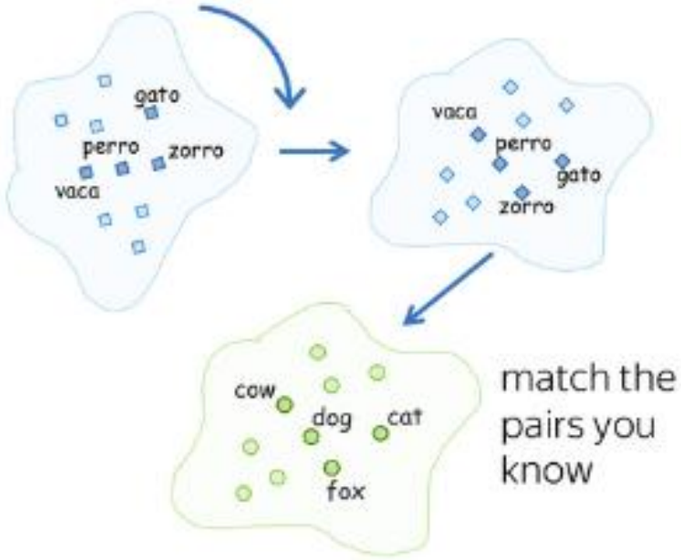
## Step 1:

- train embeddings for each language




## Step 2:

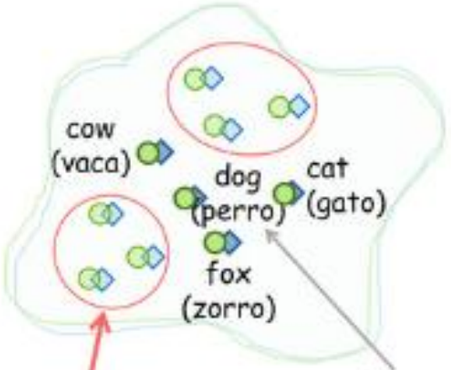
- linearly map one embeddings to the other to match words from the dictionary




match the pairs you know

## Step 3:

- after matching the two spaces, get new pairs from the new matches




old translations – the ones you knew

new translations – you learned them!

https://lena-voita.github.io/nlp_course/word_embeddings.html
https://arxiv.org/pdf/1309.4168.pdf

# Контекстные представления слов – Contextualized Word Embeddings

**недостатки предыдущих вложений – не учитывают контекст**

**«Рискую всем банком»**
**«В банке не работал кондиционер»**
**«Хранить деньги в банках не стоит»**
**«На банке сидела муха»**

**«The bank will not be accepting cash on Saturdays»**
**«The river overflowed the bank»**

**Выход:**
**языковые модели**

- embeddings in Tag LM
- CoVe
- ELMo
- Flair

# Embeddings in Tag LM

**Одна из первых работ с идеей, что недостаточно просто представлений слов**
**в задаче простановки тегов**

**Использовались представления, в котором конкатенировались 2 представления:**

- **предобученные представления слов**
- **результат предобученной нейронной LM**

Matthew E. Peters et. al. «Semi-supervised sequence tagging with bidirectional language models» // https://arxiv.org/pdf/1705.00108.pdf

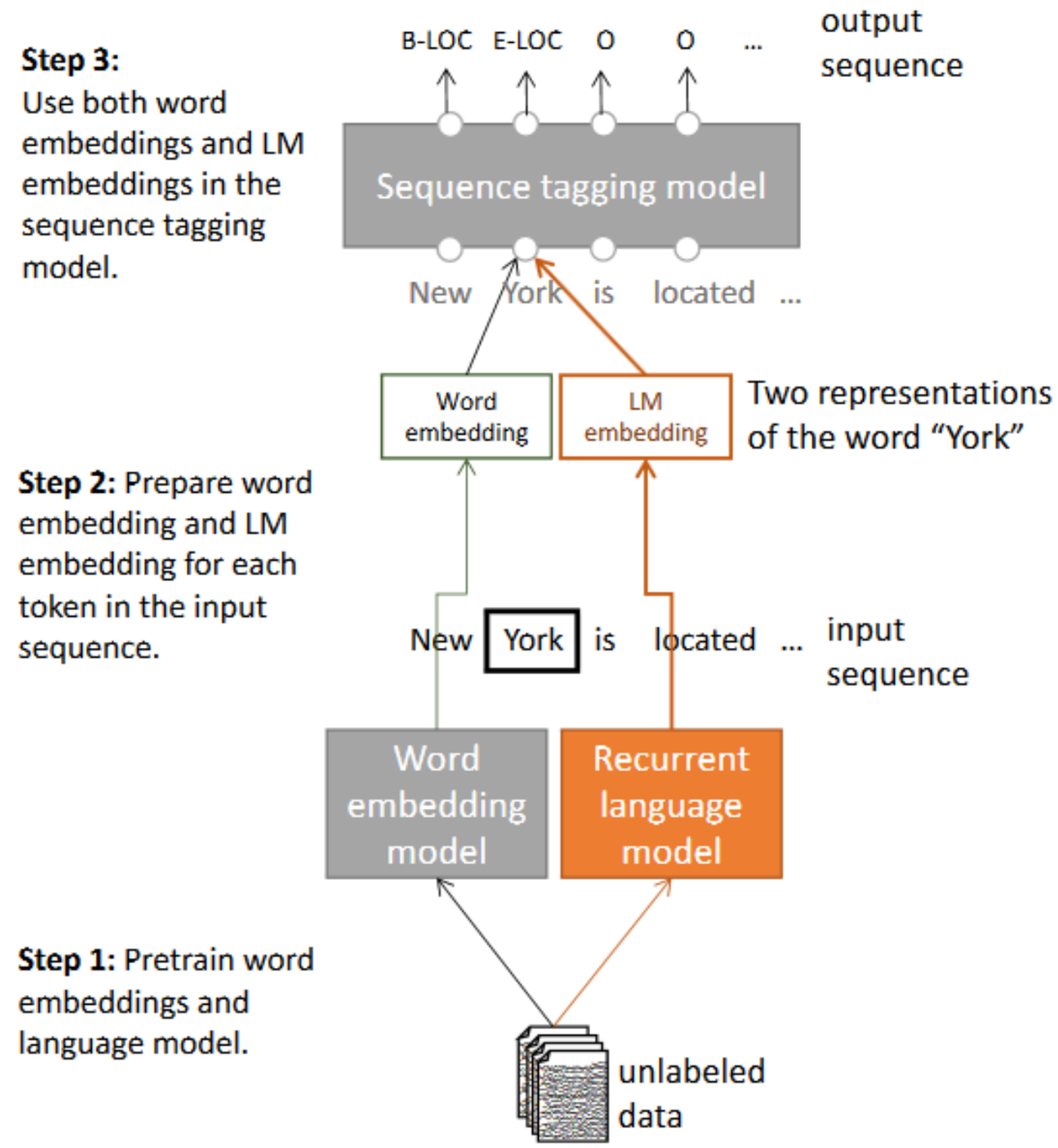
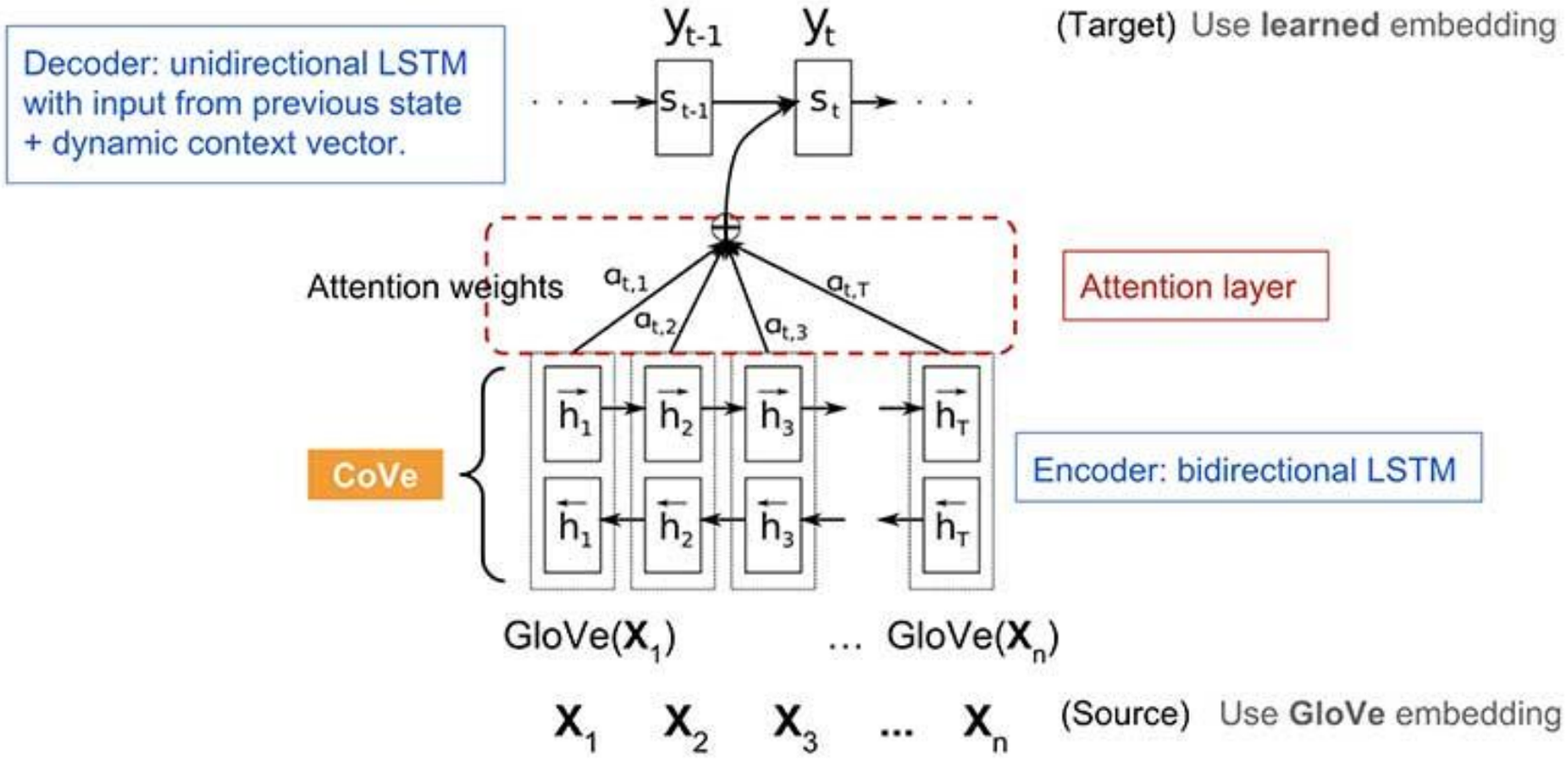



Figure 1: The main components in TagLM, our language-model-augmented sequence tagging system. The language model component (in orange) is used to augment the input token representation in a traditional sequence tagging models (in grey).

# CoVe = Contextual Word Vectors

**В отличие от классических представлений выводим кодирование слова, зависящее от контекста (всего предложений)**

**например, то что выучивает кодировщик в attentional seq-to-seq в NMT**




https://www.topbots.com/generalized-language-models-cove-elmo/

# CoVe = Contextual Word Vectors

**CoVe(x) = MT-biLSTM(GloVe(x))**

**конкатенация скрытых состояний слова [h←, h→]**

**в изначальной работе предлагалось потом в задачах классификации конкатенировать [GloVe(x), CoVe(x)]**

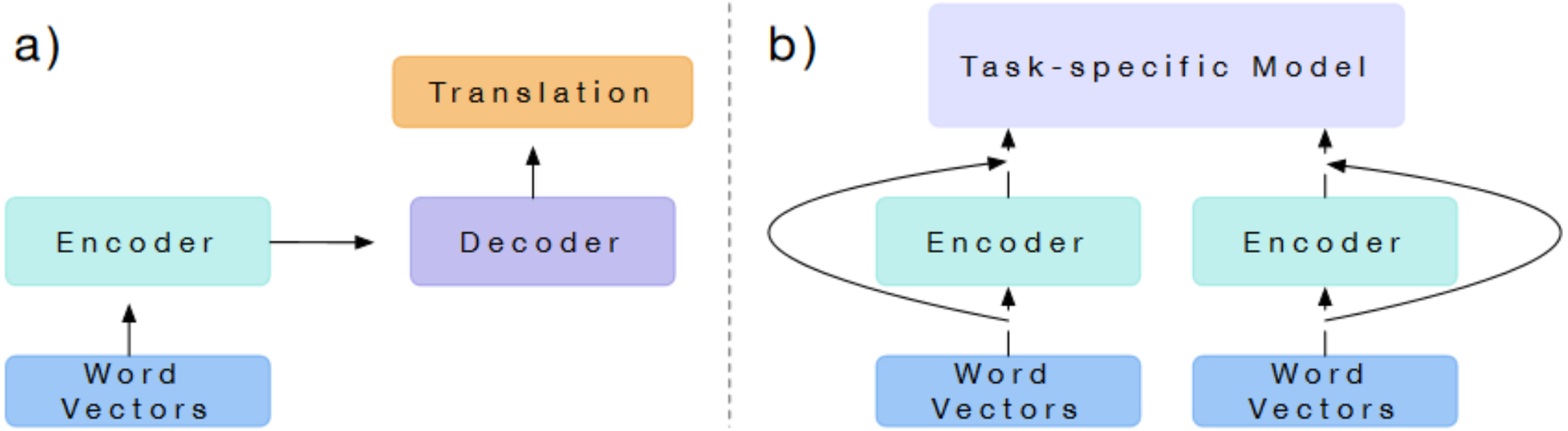
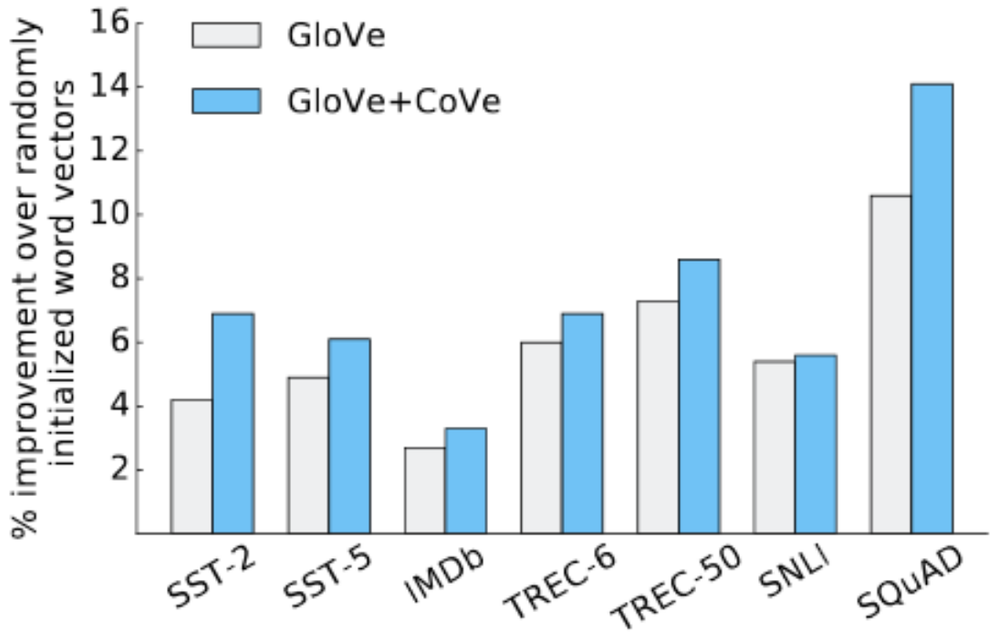


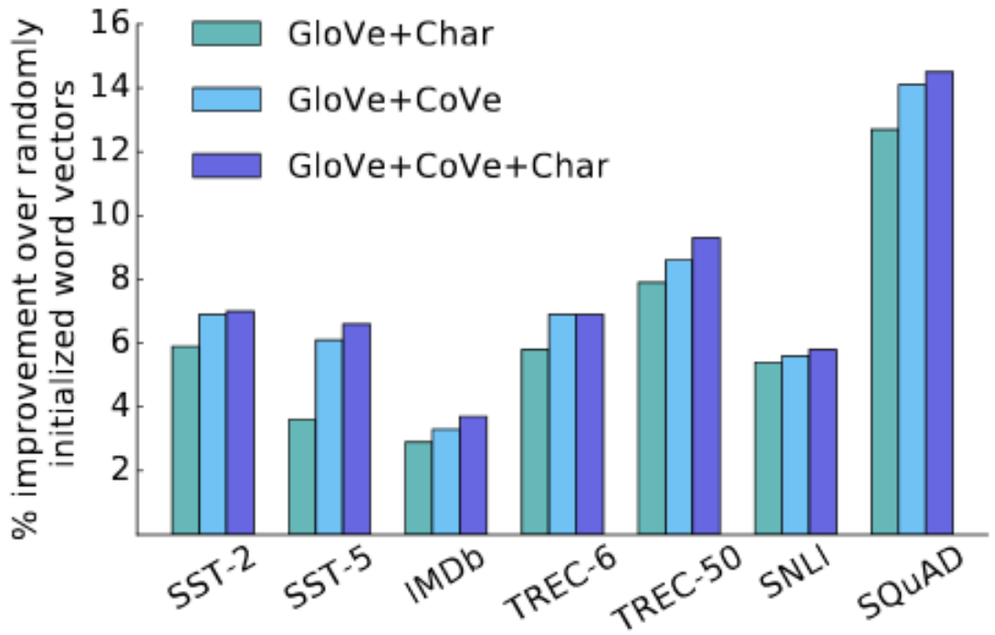

Figure 1: We a) train a two-layer, bidirectional LSTM as the encoder of an attentional sequence-to-sequence model for machine translation and b) use it to provide context for other NLP models.

**термин введён в** Bryan McCann et. al. «Learned in Translation: Contextualized Word Vectors» // https://arxiv.org/pdf/1708.00107.pdf

# CoVe = Contextual Word Vectors




(a) CoVe and GloVe




(b) CoVe and Characters

Figure 3: The Benefits of CoVe

**Char = character n-gram embeddings**

**результат не супер, как ожидалось…**

**м.б. машинный перевод более сложная задача, чем моделирование языка (что успешнее использовалось в других техниках)**

# ELMo: Embeddings from Language Models

**представление с помощью предтренировки без учителя**
**biLM обучена на большом корпусе текстов**

**новое предложение в нашей задаче пропускается через biLM**
**представление слоя = лк состояний слова**

$\Rightarrow$

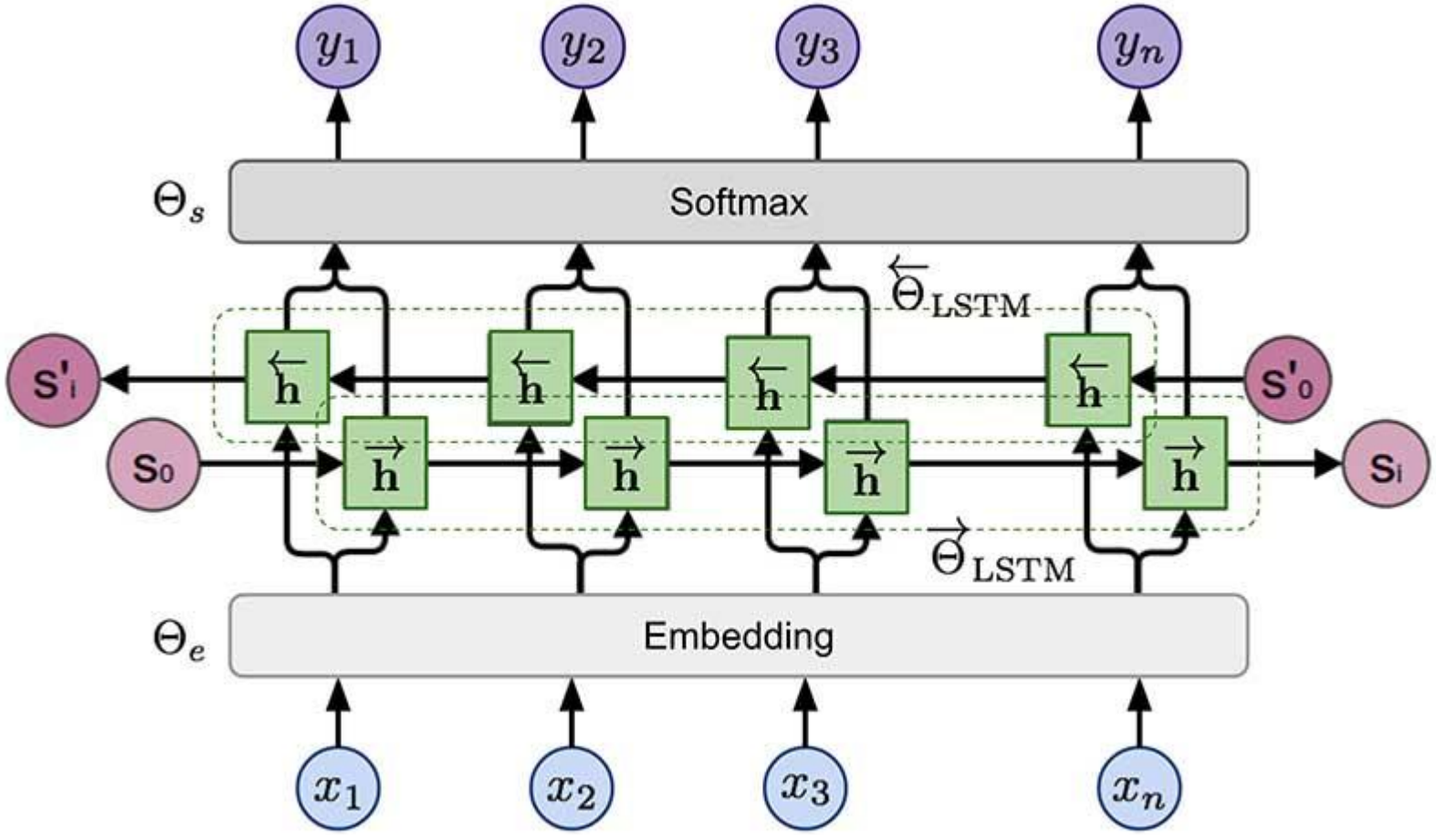
- **зависит от всего предложения**
- **глубокое (зависит от всех слоёв)**
- **есть возможность его обучать (т.к. лк)**

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, Luke Zettlemoyer «Deep contextualized word representations» // https://arxiv.org/abs/1802.05365

# ELMo: Embeddings from Language Models

**строим biLM (Bidirectional language model):**

$$\sum_k \log p(t_k \mid t_1,\ldots,t_{k-1};\Theta_x,\overrightarrow{\theta}_{\text{LSTM}},\Theta_s)) + \log p(t_k \mid t_{k+1},\ldots,t_n;\Theta_x,\overleftarrow{\theta}_{\text{LSTM}},\Theta_s))$$




$\Theta_x$ – **представление токенов**

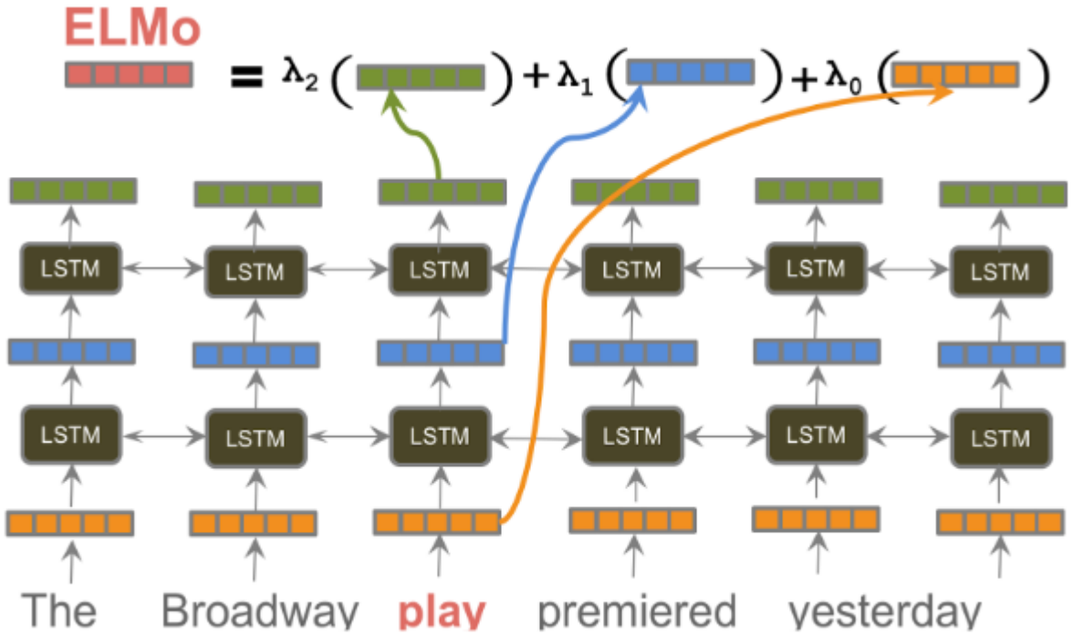
$\Theta_s$ – **softmax-слой**

https://www.topbots.com/generalized-language-models-cove-elmo/

# ELMo: Embeddings from Language Models

$$\sum_k \log p(t_k \mid t_1,\ldots,t_{k-1};\Theta_x,\overrightarrow{\theta}_{\text{LSTM}},\Theta_s)) + \log p(t_k \mid t_{k+1},\ldots,t_n;\Theta_x,\overleftarrow{\theta}_{\text{LSTM}},\Theta_s))$$

**можно затачивать представление под конкретную задачу –**
**– такую л/к скрытых состояний**




$$\text{ELMO}_k = \gamma^{\text{task}} \sum_{l\in\text{layers}} s_j^{\text{task}} [\vec{h}_{k,j}^{\text{LM}}, \overleftarrow{h}_{k,j}^{\text{LM}}]$$

**сюда ещё добавляют и выход embedding-слоя**

**разные слои – разный уровень абстракции**
**низкие ~ части речи**
**высокие ~ ответы на вопросы**

# ELMo: Embeddings from Language Models

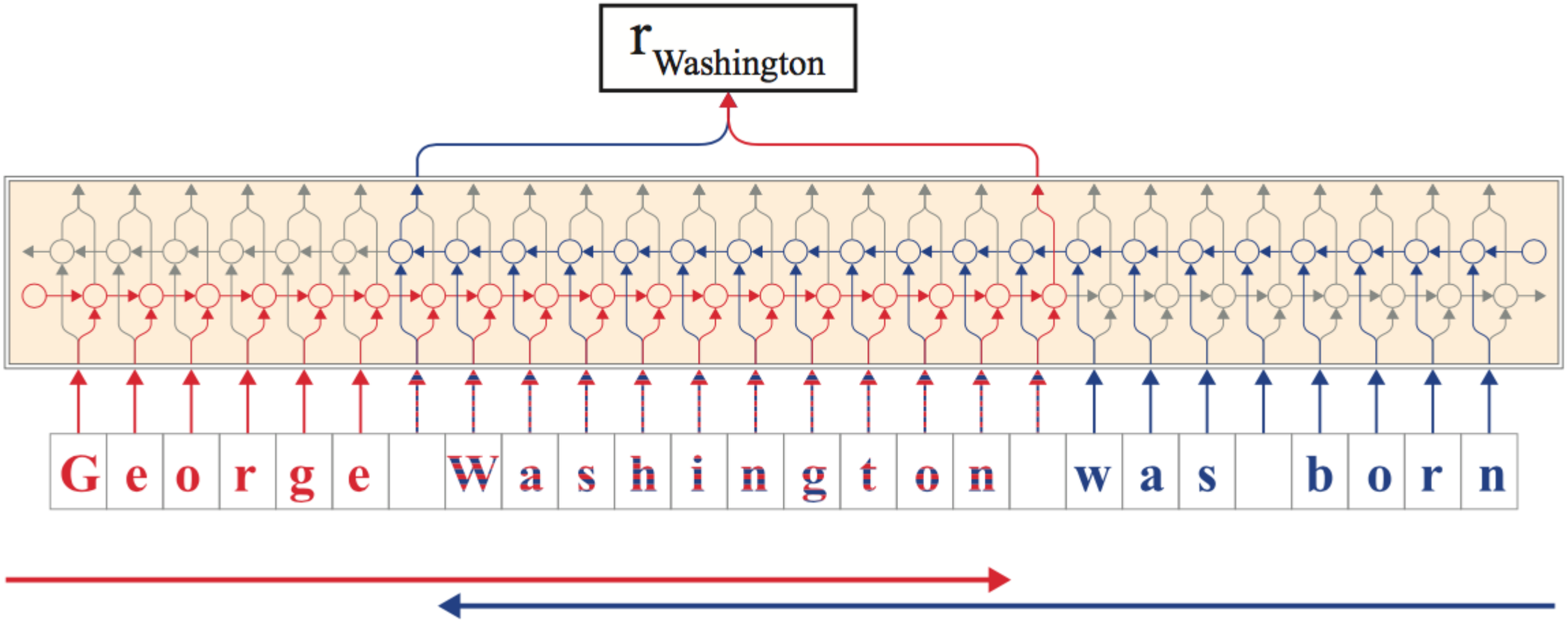
| | Source | Nearest Neighbors |
|---|---|---|
| GloVe | play | playing, game, games, played, players, plays, player, Play, football, multiplayer |
| biLM | Chico Ruiz made a spectacular play on Alusik 's grounder {...} | Kieffer , the only junior in the group , was commended for his ability to hit in the clutch , as well as his all-round excellent play . |
| | Olivia De Havilland signed to do a Broadway play for Garson {...} | {...} they were actors who had been handed fat roles in a successful play , and had talent enough to fill the roles competently , with nice understatement . |

Table 4: Nearest neighbors to "play" using GloVe and the context embeddings from a biLM.

# FLAIR: Contextual String Embeddings for Sequence Labelling

**учим посимвольную двунаправленную LM (Character-level Language Model)**

**конкатенируем скрытое состояние последней буквы LM→, первой LM←**




Alan Akbik, Duncan Blythe, Roland Vollgraf «Contextual String Embeddings for Sequence Labeling» https://www.aclweb.org/anthology/C18-1139/

# FLAIR: Contextual String Embeddings for Sequence Labelling

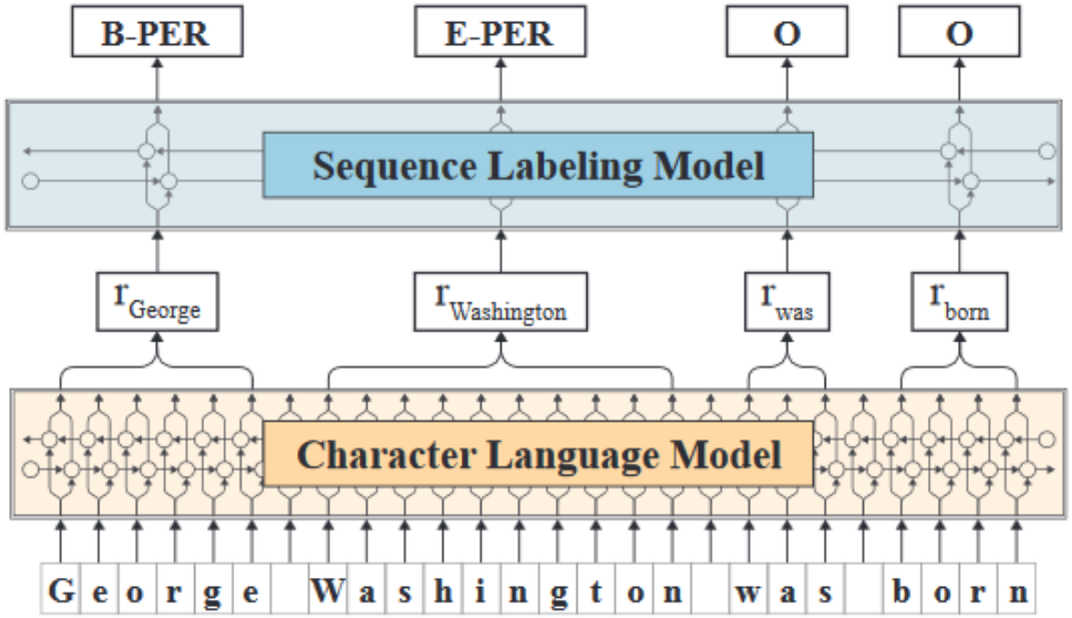



Figure 1: High level overview of proposed approach. A sentence is input as a character sequence into a pre-trained bidirectional character language model. From this LM, we retrieve for each word a contextual embedding that we pass into a vanilla BiLSTM-CRF sequence labeler, achieving robust state-of-the-art results on downstream tasks (NER in Figure).

# FLAIR: Contextual String Embeddings for Sequence Labelling

| word | context | selected nearest neighbors |
|---|---|---|
| Washington | (a) ***Washington*** *to curb support for [..]* | (1) ***Washington*** *would also take [..] action [..]*<br>(2) ***Russia*** *to clamp down on barter deals [..]*<br>(3) ***Brazil*** *to use hovercrafts for [..]* |
| Washington | (b) *[..] Anthony* ***Washington*** *(U.S.) [..]* | (1) *[..] Carla* ***Sacramento*** *( Portugal ) [..]*<br>(2) *[..] Charles* ***Austin*** *( U.S. ) [..]*<br>(3) *[..] Steve* ***Backley*** *( Britain ) [..]* |
| Washington | (c) *[..] flown to* ***Washington*** *for [..]* | (1) *[..] while visiting* ***Washington*** *to [..]*<br>(2) *[..] journey to New York City and* ***Washington*** *[..]*<br>(14) *[..] lives in* ***Chicago*** *[..]* |
| Washington | (d) *[..] when* ***Washington*** *came charging back [..]* | (1) *[..] point for victory when* ***Washington*** *found [..]*<br>(4) *[..] before* ***England*** *struck back with [..]*<br>(6) *[..] before* ***Ethiopia*** *won the spot kick decider [..]* |
| Washington | (e) *[..] said* ***Washington*** *[..]* | (1) *[..] subdue the never-say-die* ***Washington*** *[..]*<br>(4) *[..] a private school in* ***Washington*** *[..]*<br>(9) *[..] said* ***Florida*** *manager John Boles [..]* |

Table 4: Examples of the word "Washington" in different contexts in the CoNLL03 data set, and nearest neighbors using cosine distance over our proposed embeddings. Since our approach produces different embeddings based on context, we retrieve different nearest neighbors for each mention of the same word.

# Совместное использование представлений

можно конкатенировать разные представления

использовать одни как инициализации для вычисления других

# Другие решения

**BERT – будет дальше**

**не просто контекст слева и справа, а сразу анализирует всё!**

**Раньше**

**Кот сидел на крыше около трубы**

**Потом**

**Кот сидел на крыше около трубы**

# Представление текстов

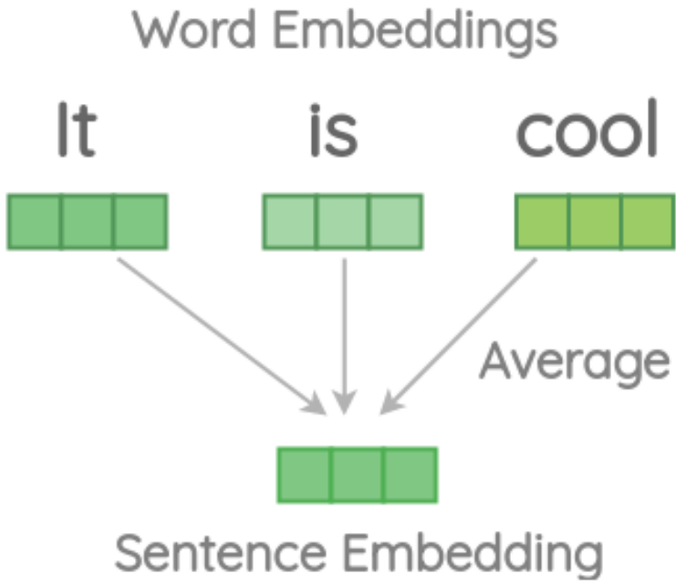
**умеем представлять (вкладывать) слова**
**как быть с предложениями / абзацами / текстами?**


https://amitness.com/2020/06/universal-sentence-encoder/

# Представление текстов

**текст ~ «среднее» векторов входящих слов**

**~ сумма с весами – вероятностями слов**




**уже было в seq2seq**

**– нет учёта порядка слов**

**– «It» ~ «It is cool» похожи (0.89)**

# Представление текстов

## Универсальный подход, когда есть сходство / различие пар

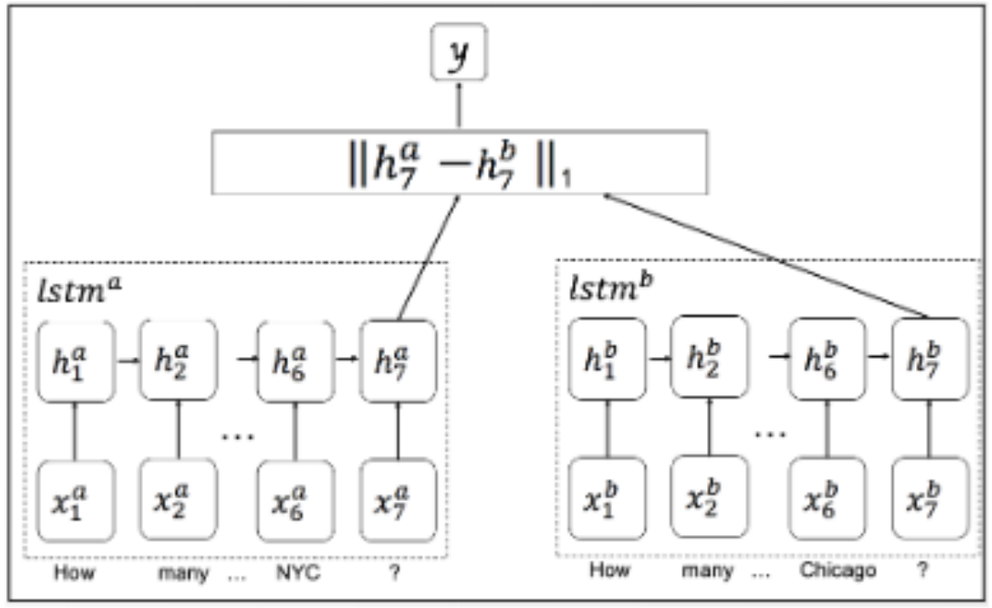



Fig. 1. Siamese LSTM Architecture

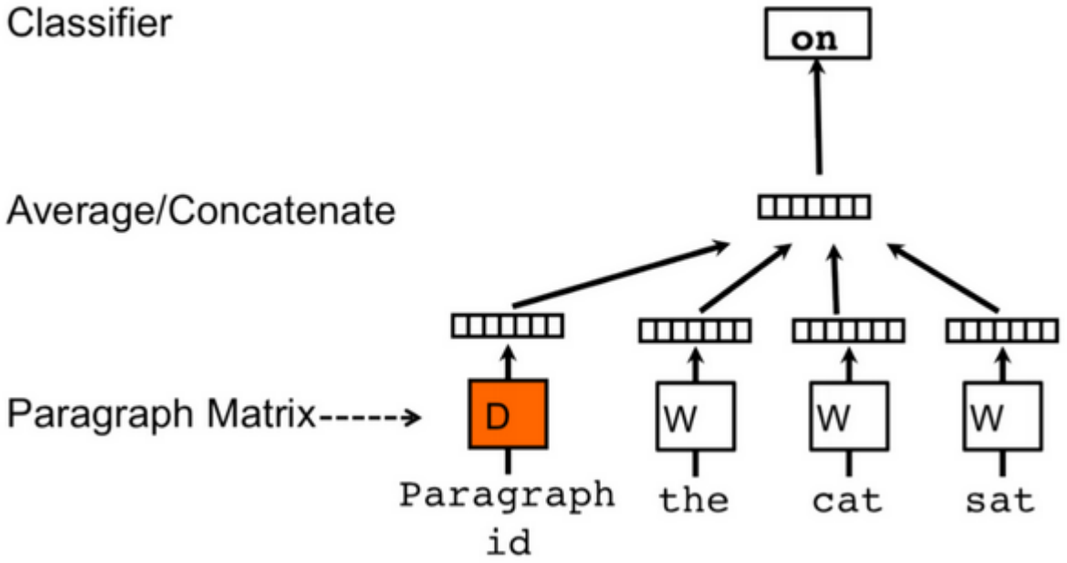
Fig. 5. Siamese LSTM representation for the Most frequent types of SQL queries

Yassine Benajiba et al. «Siamese Networks for Semantic Pattern Similarity» // https://arxiv.org/pdf/1812.06604.pdf

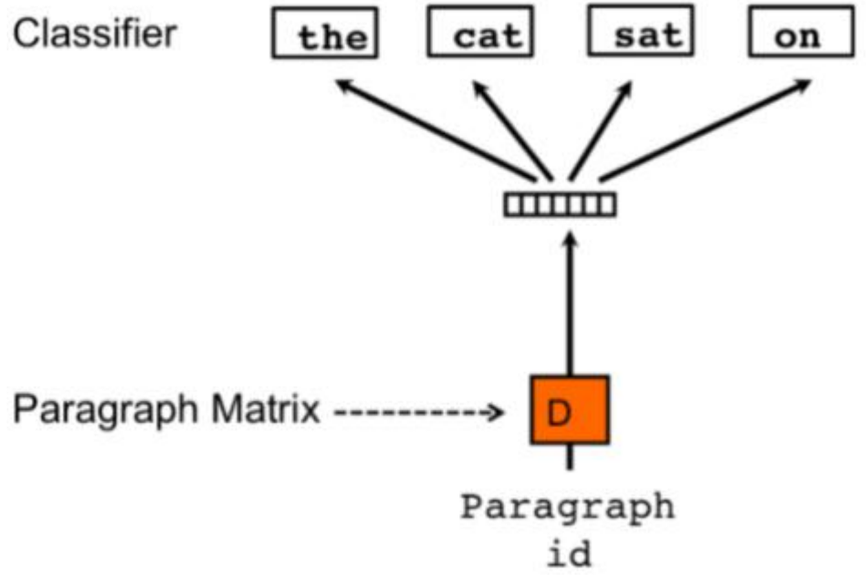
# Представление текстов: Paragraph Vector (Doc2Vec / paragraph2vec)

## По аналогии с word2vec

### PV-DM (Distributed Memory)




### Distributed Bag Of Words (DBOW)




предсказываем случайно выбранные слова

Quoc V. Le, Tomas Mikolov Distributed Representations of Sentences and Documents // https://arxiv.org/abs/1405.4053

# Представление предложений: The skip-thoughts model

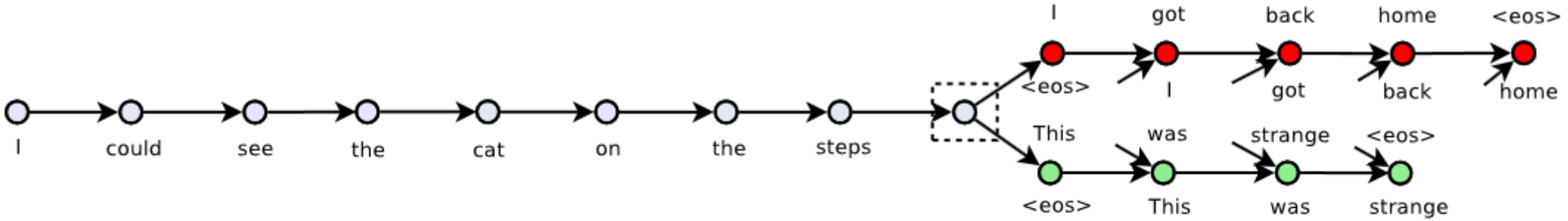



Figure 1: The skip-thoughts model. Given a tuple $(s_{i-1}, s_i, s_{i+1})$ of contiguous sentences, with $s_i$ the $i$-th sentence of a book, the sentence $s_i$ is encoded and tries to reconstruct the previous sentence $s_{i-1}$ and next sentence $s_{i+1}$. In this example, the input is the sentence triplet *I got back home. I could see the cat on the steps. This was strange.* Unattached arrows are connected to the encoder output. Colors indicate which components share parameters. ⟨eos⟩ is the end of sentence token.

**Последовательность предложений:**

**I got back home. I could see the cat on the steps. This was strange.**

**пытаемся по среднему предсказать первое и третье**

**один цвет – разделение параметров**

$$\sum_t \log P(w_{i+1}^t | w_{i+1}^{<t}, \mathbf{h}_i) + \sum_t \log P(w_{i-1}^t | w_{i-1}^{<t}, \mathbf{h}_i)$$

**кодировщик-декодировщик**

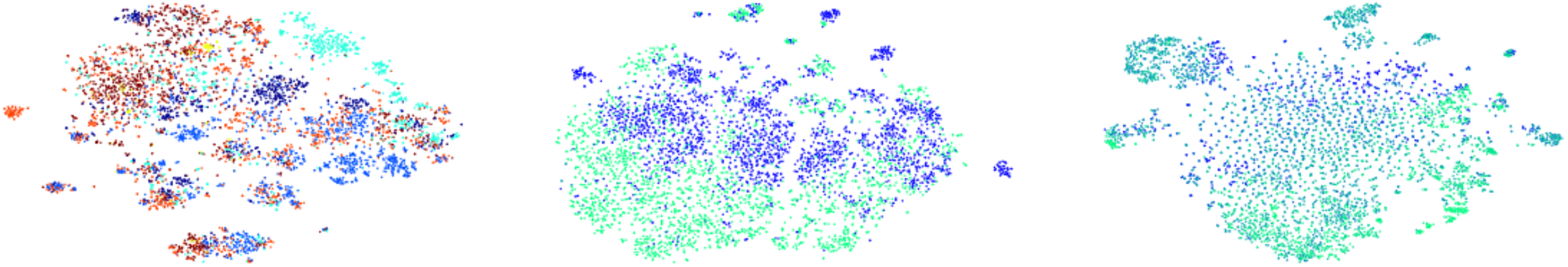
**довольно долгий, но качество высокое**

# The skip-thoughts model: ближайшие соседи

| Query and nearest sentence |
|---|
| he ran his hand inside his coat , double-checking that the unopened letter was still there .<br>he slipped his hand between his coat and his shirt , where the folded copies lay in a brown envelope . |
| im sure youll have a glamorous evening , she said , giving an exaggerated wink .<br>im really glad you came to the party tonight , he said , turning to her . |
| although she could tell he had n't been too invested in any of their other chitchat , he seemed genuinely curious about this .<br>although he had n't been following her career with a microscope , he 'd definitely taken notice of her appearances . |
| an annoying buzz started to ring in my ears , becoming louder and louder as my vision began to swim .<br>a weighty pressure landed on my lungs and my vision blurred at the edges , threatening my consciousness altogether . |
| if he had a weapon , he could maybe take out their last imp , and then beat up errol and vanessa .<br>if he could ram them from behind , send them sailing over the far side of the levee , he had a chance of stopping them . |
| then , with a stroke of luck , they saw the pair head together towards the portaloos .<br>then , from out back of the house , they heard a horse scream probably in answer to a pair of sharp spurs digging deep into its |

Ryan Kiros, et al. «Skip-Thought Vectors» // https://arxiv.org/abs/1506.06726

# The skip-thoughts model: ближайшие соседи



(a) TREC (b) SUBJ (c) SICK

Figure 2: t-SNE embeddings of skip-thought vectors on different datasets. Points are colored based on their labels (question type for TREC, subjectivity/objectivity for SUBJ). On the SICK dataset, each point represents a sentence pair and points are colored on a gradient based on their relatedness labels. Results best seen in electronic form.

# Предтренировка автокодировщика (Autoencoder pretraining)

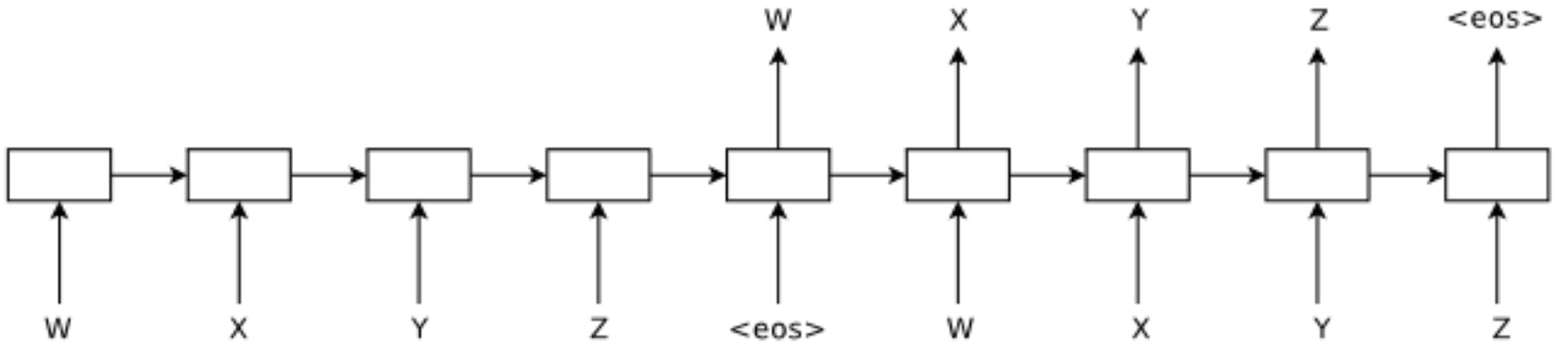



Figure 1: The sequence autoencoder for the sequence "WXYZ". The sequence autoencoder uses a recurrent network to read the input sequence in to the hidden state, which can then be used to reconstruct the original sequence.

**хотим, чтобы автокодировщик воспроизводил входную последовательность!**

Andrew M. Dai, Quoc V. Le «Semi-supervised Sequence Learning» //
https://arxiv.org/abs/1511.01432

# Представление слов/предложений/текстов: StarSpace

**название: $*$ → «space» (пространтсво)**

**Метод оперирует с объектами, которые описываются наборами признаков из фиксированного множества (представление объекта = сумма представлений)**

**Пример: предложение = набор слов**

**(или = набор n-грамм)**

$$\sum_{\substack{(a,b)\in K^+ \\ b^-\in K^-}} L^{\text{batch}}(\text{sim}(a,b),\text{sim}(a,b_1^-),\ldots,\text{sim}(a,b_k^-)) \to \min$$

**~ есть генератор позитивных и негативных пар**

**под решение конкретной задачи**

**представление всех сущностей (документы, картинки и т.п.) в едином пространстве**

Ledell Wu, Adam Fisch, Sumit Chopra, Keith Adams, Antoine Bordes, Jason Weston StarSpace: Embed All The Things! //
https://arxiv.org/abs/1709.03856
https://github.com/facebookresearch/StarSpace

# Представление слов/предложений/текстов: StarSpace

| Input Query | StarSpace result | fastText result |
|---|---|---|
| She is the 1962 Blue Swords champion and 1960 Winter Universiade silver medalist. | **Article**: Eva Groajov.<br>**Paragraph**: Eva Groajov , later Bergerov-Groajov , is a former competitive figure skater who represented Czechoslovakia. She placed 7th at the 1961 European Championships and 13th at the 1962 World Championships. She was coached by Hilda Mdra. | **Article**: Michael Reusch.<br>**Paragraph**: Michael Reusch (February 3, 1914April 6 , 1989) was a Swiss gymnast and Olympic Champion. He competed at the 1936 Summer Olympics in Berlin, where he received silver medals in parallel bars and team combined exercises... |
| The islands are accessible by a one-hour speedboat journey from Kuala Abai jetty, Kota Belud, 80 km north-east of Kota Kinabalu, the capital of Sabah. | **Article**: Mantanani Islands.<br>**Paragraph**: The Mantanani Islands form a small group of three islands off the north-west coast of the state of Sabah, Malaysia, opposite the town of Kota Belud, in northern Borneo. The largest island is Mantanani Besar; the other two are Mantanani Kecil and Lungisan... | **Article**: Gum-Gum<br>**Paragraph**: Gum-Gum is a township of Sandakan, Sabah, Malaysia. It is situated about 25km from Sandakan town along Labuk Road. |
| Maggie withholds her conversation with Neil from Tom and goes to the meeting herself, and Neil tells her the spirit that contacted Tom has asked for something and will grow upset if it does not get done. | **Article**: Stir of Echoes<br>**Paragraph**: Stir of Echoes is a 1999 American supernatural horror-thriller released in the United States on September 10 , 1999 , starring Kevin Bacon and directed by David Koepp . The film is loosely based on the novel ”A Stir of Echoes” by Richard Matheson... | **Article**: The Fabulous Five<br>**Paragraph**: The Fabulous Five is an American book series by Betsy Haynes in the late 1980s . Written mainly for preteen girls , it is a spin-off of Haynes ’ other series about Taffy Sinclair... |

Table 8: StarSpace predictions for some example Wikipedia Article Search (Task 1) queries where StarSpace is correct.

# Представление слов/предложений/текстов: StarSpace

| Task | MR | CR | SUBJ | MPQA | SST | TREC | MRPC | SICK-R | SICK-E | STS14 |
|---|---|---|---|---|---|---|---|---|---|---|
| Unigram-TFIDF* | 73.7 | 79.2 | 90.3 | 82.4 | - | 85.0 | 73.6 / 81.7 | - | - | 0.58 / 0.57 |
| ParagraphVec (DBOW)* | 60.2 | 66.9 | 76.3 | 70.7 | - | 59.4 | 72.9 / 81.1 | - | - | 0.42 / 0.43 |
| SDAE* | 74.6 | 78.0 | 90.8 | 86.9 | - | 78.4 | 73.7 / 80.7 | - | - | 0.37 / 0.38 |
| SIF(GloVe+WR)* | - | - | - | 82.2 | - | - | - | - | 84.6 | 0.69 / - |
| word2vec* | 77.7 | 79.8 | 90.9 | 88.3 | 79.7 | 83.6 | 72.5 / 81.4 | 0.80 | 78.7 | 0.65 / 0.64 |
| GloVe* | 78.7 | 78.5 | 91.6 | 87.6 | 79.8 | 83.6 | 72.1 / 80.9 | 0.80 | 78.6 | 0.54 / 0.56 |
| fastText (public Wikipedia model)* | 76.5 | 78.9 | 91.6 | 87.4 | 78.8 | 81.8 | 72.4 / 81.2 | 0.80 | 77.9 | 0.63 / 0.62 |
| StarSpace [word] | 73.8 | 77.5 | 91.53 | 86.6 | 77.2 | 82.2 | 73.1 / 81.8 | 0.79 | 78.8 | 0.65 / 0.62 |
| StarSpace [sentence] | 69.1 | 75.1 | 85.4 | 80.5 | 72.0 | 63.0 | 69.2 / 79.7 | 0.76 | 76.2 | 0.70 / 0.67 |
| StarSpace [word + sentence] | 72.1 | 77.1 | 89.6 | 84.1 | 77.5 | 79.0 | 70.2 80.3 | 0.79 | 77.8 | 0.69/0.66 |
| StarSpace [ensemble w+s] | 76.6 | 80.3 | 91.8 | 88.0 | 79.9 | 85.2 | 71.8 / 80.6 | 0.78 | 82.1 | 0.69 / 0.65 |

Table 9: Transfer test results on SentEval. * indicates model results that have been extracted from (Conneau et al. 2017). For MR, CR, SUBJ, MPQA, SST, TREC, SICK-R we report accuracies; for MRPC, we report accuracy/F1; for SICK-R we report Pearson correlation with relatedness score; for STS we report Pearson/Spearman correlations between the cosine distance of two sentences and human-labeled similarity score.

# Представление предложений: Deep Averaging Network (DAN)

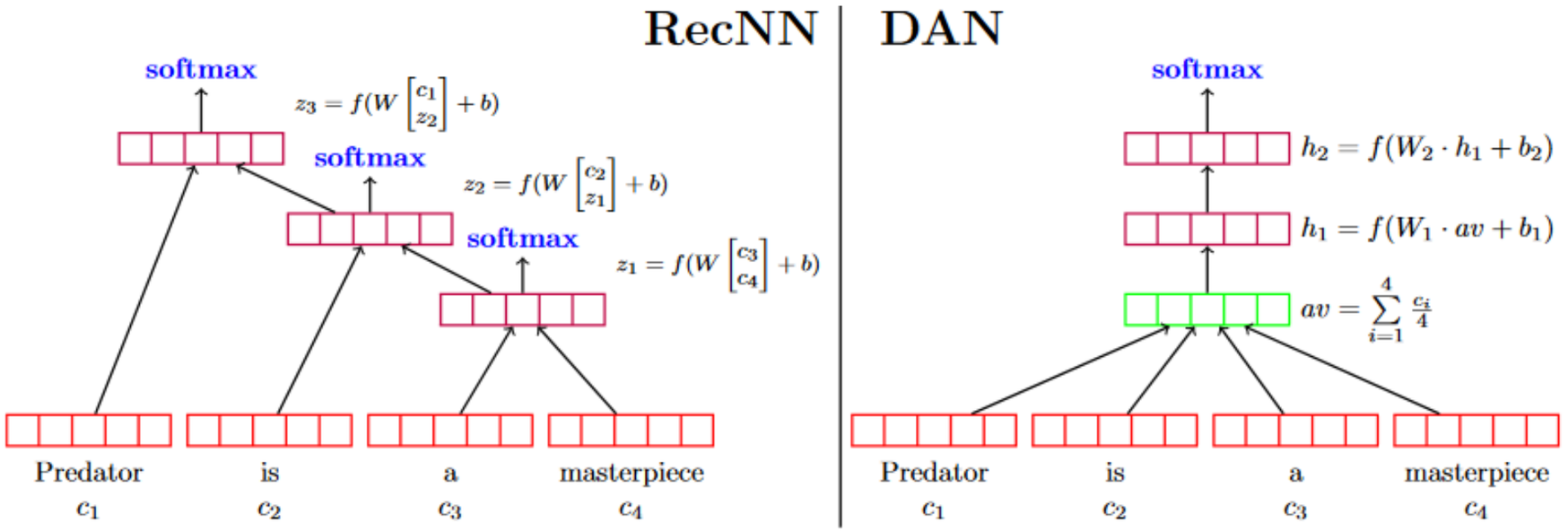



Figure 1: On the left, a **RecNN** is given an input sentence for sentiment classification. Softmax layers are placed above every internal node to avoid vanishing gradient issues. On the right is a two-layer **DAN** taking the same input. While the **RecNN** has to compute a nonlinear representation (purple vectors) for every node in the parse tree of its input, this **DAN** only computes two nonlinear layers for every possible input.

**Простое усреднение…**

**Подумать – по сути это классификация**

M. Iyyer, etc. Deep Unordered Composition Rivals Syntactic Methods for Text Classification, 2015 // http://www.aclweb.org/anthology/P15-1162

# Представление предложений: Deep Averaging Network (DAN)

1. **Task**: map an input sequence of tokens $X$ to one of $k$ labels
2. **Composition** function $g$ averages word embeddings:

$$z = g(w \in X) = \frac{1}{|X|} \sum_{w \in X} v_w,$$

where $v_w$ is a word embedding of word $w$

3. Estimate **probabilities** for each output label:
$\hat{y} = \text{softmax}(W_s \times z + b)$ and **predict** the label with highest probability
4. **Training**: minimize cross-entropy error: $\sum_{p=1}^{k} y_p \log \hat{y}_p$

Add more layers:

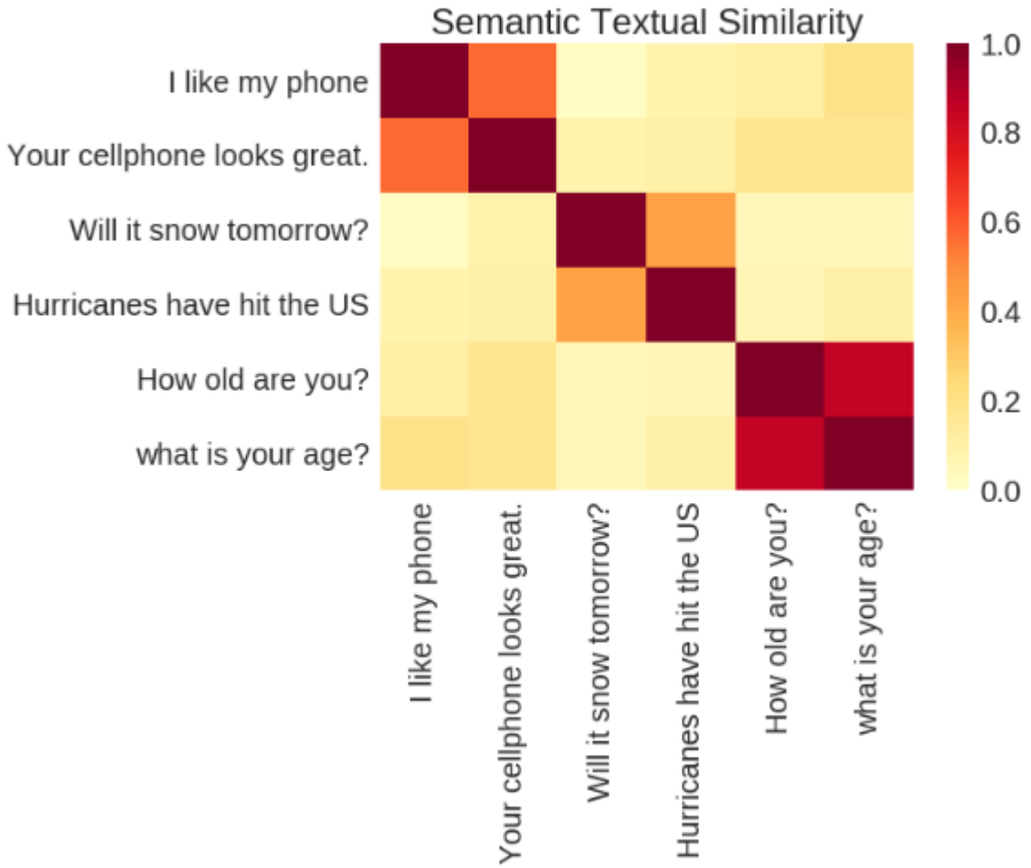
$$z_i = g(z_{i-1}) = f(W_i \times z_{i-1} + b_i)$$

**Word dropout**: drop word tokens' entire word embeddings from the vector average

| Sentence | DAN | DRecNN | Ground Truth |
|---|---|---|---|
| a lousy movie that's not merely unwatchable, but also unlistenable | negative | negative | negative |
| if you're not a prepubescent girl, you'll be laughing at britney spears' movie-starring debut whenever it does n't have you impatiently squinting at your watch | negative | negative | negative |
| blessed with immense physical prowess he may well be, but ahola is simply not an actor | positive | neutral | negative |
| who knows what exactly godard is on about in this film, but his words and images do n't have to add up to mesmerize you. | positive | positive | positive |
| it's so good that its relentless, polished wit can withstand not only inept school productions, but even oliver parker's movie adaptation | negative | positive | positive |
| too bad, but thanks to some lovely comedic moments and several fine performances, it's not a total loss | negative | negative | positive |
| this movie was not good | negative | negative | negative |
| this movie was good | positive | positive | positive |
| this movie was bad | negative | negative | negative |
| the movie was not bad | negative | negative | positive |

Table 3: Predictions of DAN and DRecNN models on real (top) and synthetic (bottom) sentences that contain negations and contrastive conjunctions. In the first column, words colored red individually predict the negative label when fed to a DAN, while blue words predict positive. The DAN learns that the negators *not* and *n't* are strong negative predictors, which means it is unable to capture double negation as in the last real example and the last synthetic example. The DRecNN does slightly better on the synthetic double negation, predicting a lower negative polarity.

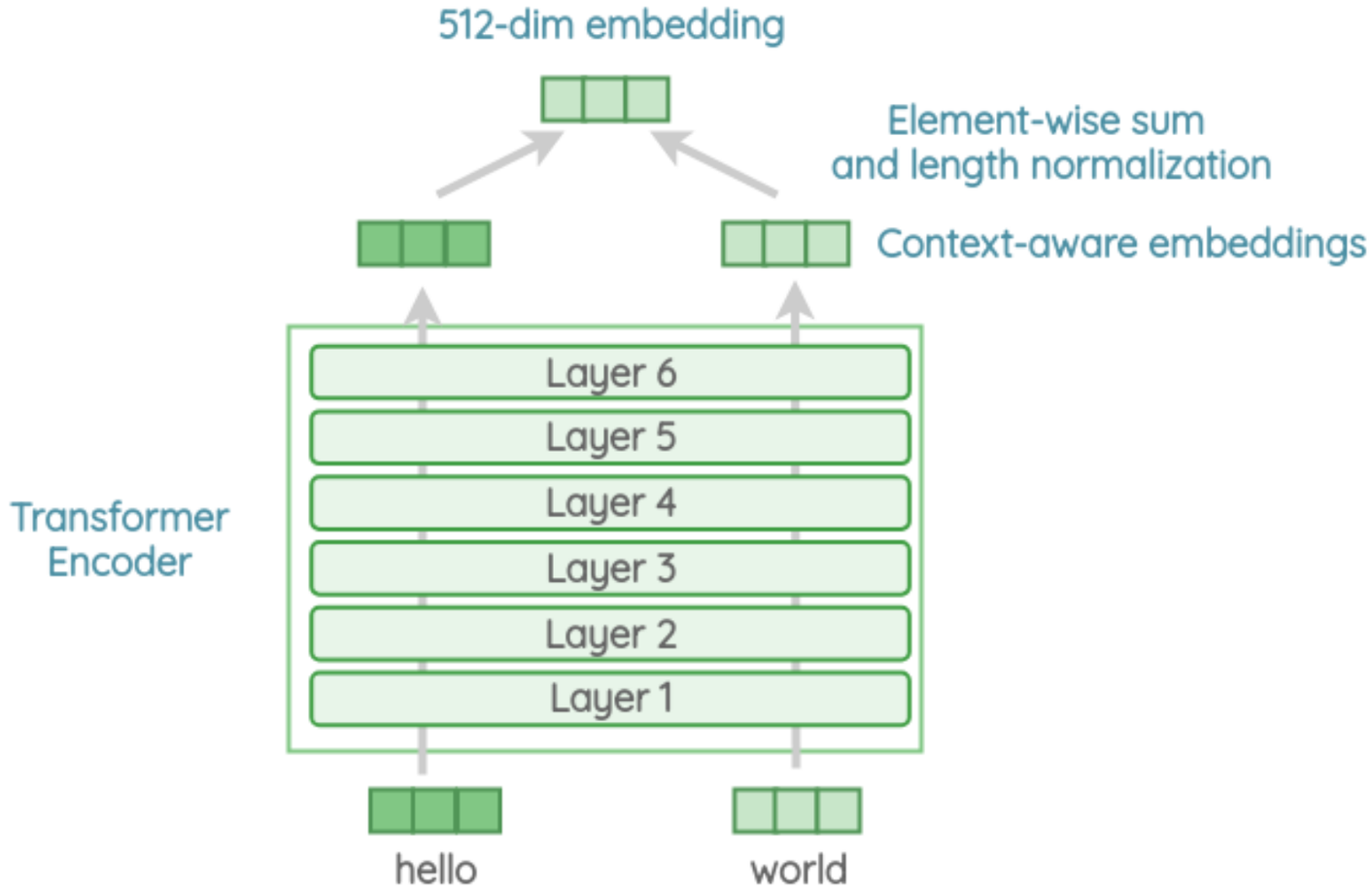
# Universal Sentence Encoder




**использовали 1) Transformer (чуть лучше) 2) DAN**

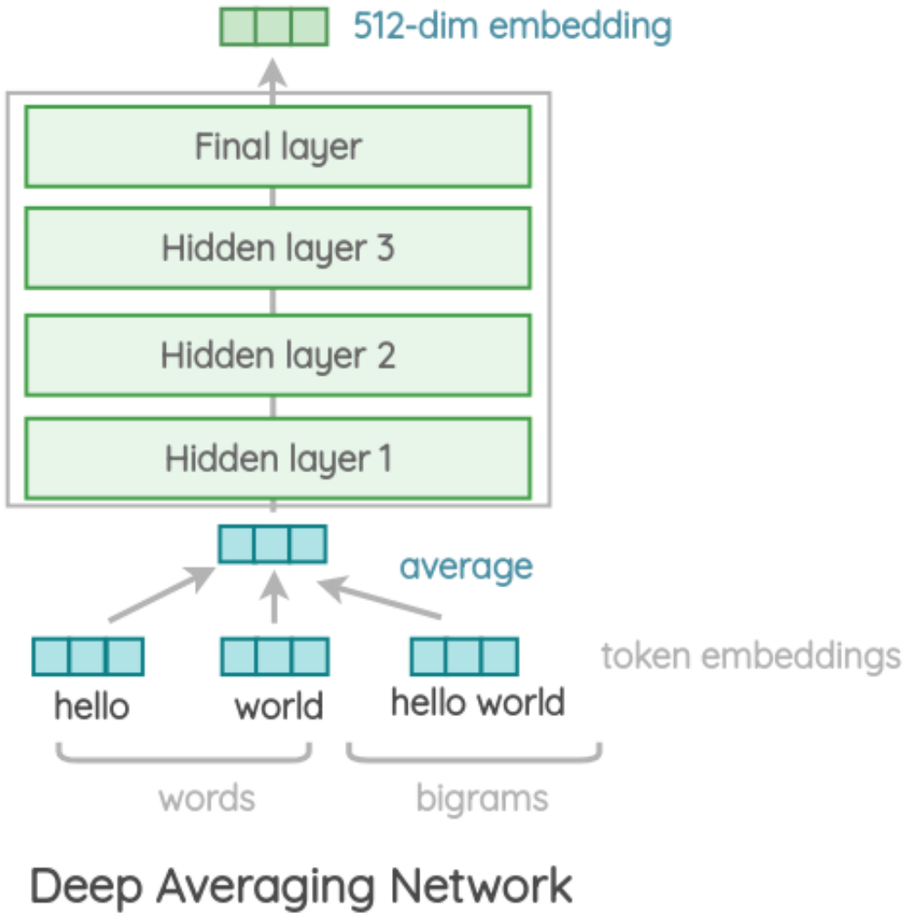
Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, Ray Kurzweil Universal Sentence Encoder // https://arxiv.org/abs/1803.11175

# Universal Sentence Encoder: Transformer




https://amitness.com/2020/06/universal-sentence-encoder/

# Universal Sentence Encoder: DAN

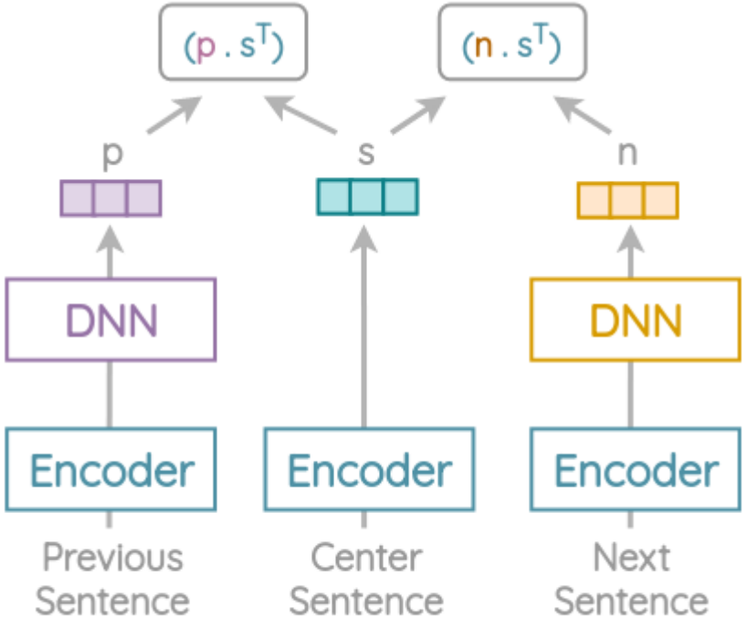



**тут ещё биграммы используются**

# Universal Sentence Encoder: Multi-task Learning

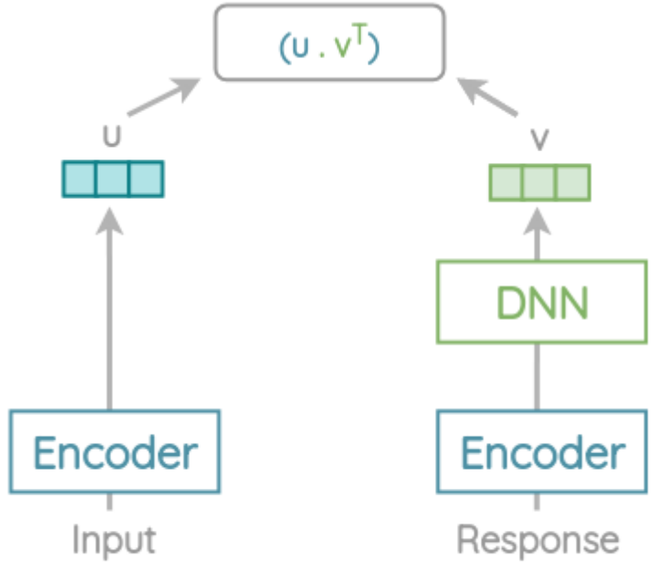
## Обучаем представление затачиваясь на несколько задач

**1) Modified Skip-thought – предсказываем предыдущее и следующее**




Skip-thought Task Structure

**2) Conversational Input-Response Prediction – предсказать ответ среди перечня (есть правильный и случайные)**

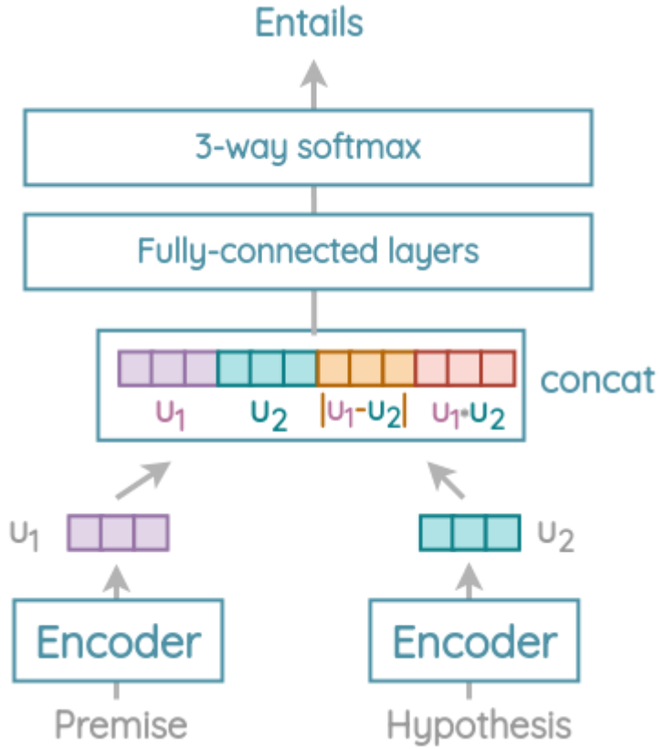



Input-Response Prediction

# Universal Sentence Encoder: Multi-task Learning

## 3) Natural Language Inference – есть ли противоречие или следствие

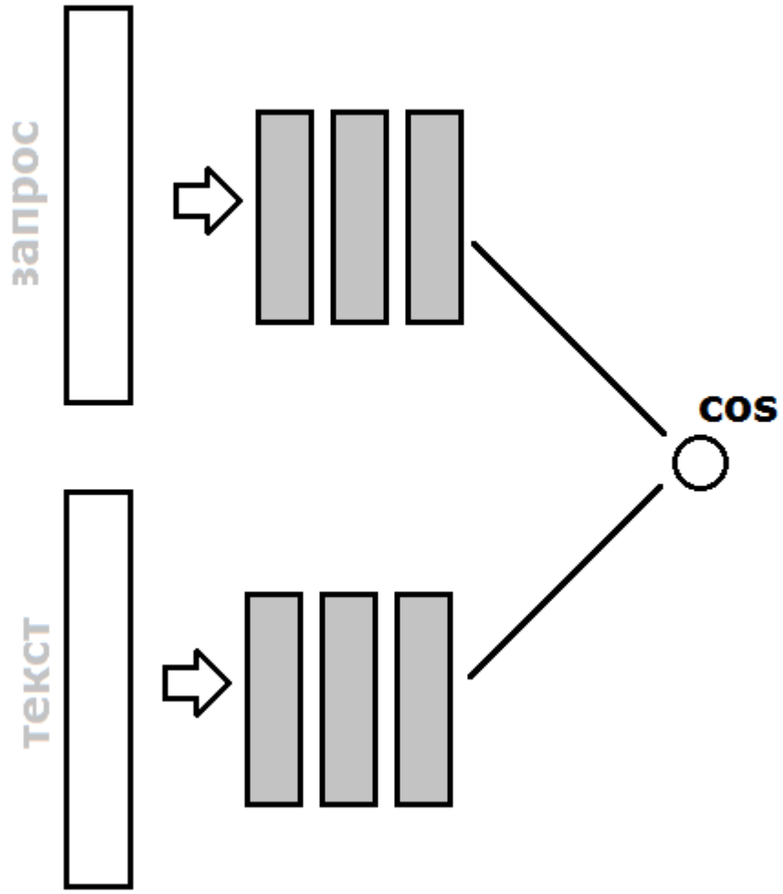
| Premise | Hypothesis | Judgement |
|---|---|---|
| A soccer game with multiple males playing | Some men are playing a sport | entailment |
| I love Marvel movies | I hate Marvel movies | contradiction |
| I love Marvel movies | A ship arrived | neutral |



| Model | MR | CR | SUBJ | MPQA | TREC | SST | STS Bench (dev / test) |
|---|---|---|---|---|---|---|---|
| Sentence & Word Embedding Transfer Learning | | | | | | | |
| USE_D+DAN (w2v w.e.) | 77.11 | 81.71 | 93.12 | 87.01 | 94.72 | 82.14 | – |
| USE_D+CNN (w2v w.e.) | 78.20 | 82.04 | 93.24 | 85.87 | 97.67 | 85.29 | – |
| USE_T+DAN (w2v w.e.) | 81.32 | 86.66 | 93.90 | 88.14 | 95.51 | 86.62 | – |
| USE_T+CNN (w2v w.e.) | 81.18 | 87.45 | 93.58 | 87.32 | 98.07 | 86.69 | – |
| *Sentence Embedding Transfer Learning* | | | | | | | |
| USE_D | 74.45 | 80.97 | 92.65 | 85.38 | 91.19 | 77.62 | 0.763 / 0.719 (r) |
| USE_T | 81.44 | 87.43 | 93.87 | 86.98 | 92.51 | 85.38 | 0.814 / 0.782 (r) |
| USE_D+DAN (lrn w.e.) | 77.57 | 81.93 | 92.91 | 85.97 | 95.86 | 83.41 | – |
| USE_D+CNN (lrn w.e.) | 78.49 | 81.49 | 92.99 | 85.53 | 97.71 | 85.27 | – |
| USE_T+DAN (lrn w.e.) | 81.36 | 86.08 | 93.66 | 87.14 | 96.60 | 86.24 | – |
| USE_T+CNN (lrn w.e.) | 81.59 | 86.45 | 93.36 | 86.85 | 97.44 | 87.21 | – |
| *Word Embedding Transfer Learning* | | | | | | | |
| DAN (w2v w.e.) | 74.75 | 75.24 | 90.80 | 81.25 | 85.69 | 80.24 | – |
| CNN (w2v w.e.) | 75.10 | 80.18 | 90.84 | 81.38 | 97.32 | 83.74 | – |
| *Baselines with No Transfer Learning* | | | | | | | |
| DAN (lrn w.e.) | 75.97 | 76.91 | 89.49 | 80.93 | 93.88 | 81.52 | – |
| CNN (lrn w.e.) | 76.39 | 79.39 | 91.18 | 82.20 | 95.82 | 84.90 | – |

Table 2: Model performance on transfer tasks. *USE_T* is the universal sentence encoder (USE) using Transformer. *USE_D* is the universal encoder DAN model. Models tagged with *w2v w.e.* make use of pre-training word2vec skip-gram embeddings for the transfer task model, while models tagged with *lrn w.e.* use randomly initialized word embeddings that are learned only on the transfer task data. Accuracy is reported for all evaluations except STS Bench where we report the Pearson correlation of the similarity scores with human judgments. Pairwise similarity scores are computed directly using the sentence embeddings from the universal sentence encoder as in Eq. (1).

# DSSM = Deep Structured Semantic Model




**сиамская сеть**

# DSSM = Deep Structured Semantic Model

https://www.researchgate.net/publication/262289160_Learning_deep_structured_semantic_models_for_web_search_using_clickthrough_data

**вход – не только слова, но и n-граммы (вместе с ними – конкатенация)**

https://habr.com/company/yandex/blog/314222

**часто легко найти положительные примеры**

**отрицательные**

**1) берутся случайные – обучаются сети**

**2) берутся те, у которых высокая вероятность класса +, но они –**

**3) повторяется п. 2**

# InferSent – Supervised sentence embedding

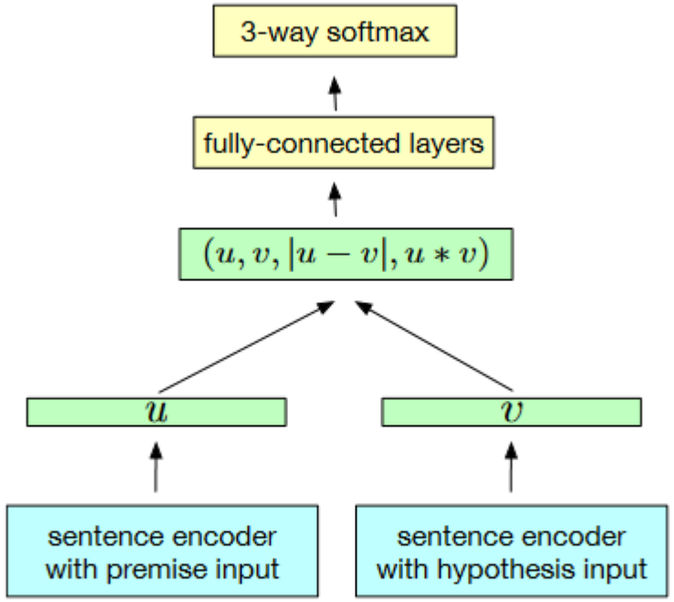



Figure 1: **Generic NLI training scheme.**

**Обучаем предсказывая метки «entailment», «neutral», «contradictive»**

**570k предложний, сгенерированных и размеченных людьми**

**cross-entropy**

**Проверили 7 разных архитектур**

Alexis Conneau et al. «Supervised Learning of Universal Sentence Representations from Natural Language Inference Data» // https://arxiv.org/pdf/1705.02364.pdf

# InferSent – Supervised sentence embedding

| Model | dim | NLI dev | NLI test | Transfer micro | Transfer macro |
|---|---|---|---|---|---|
| LSTM | 2048 | 81.9 | 80.7 | 79.5 | 78.6 |
| GRU | 4096 | 82.4 | 81.8 | 81.7 | 80.9 |
| BiGRU-last | 4096 | 81.3 | 80.9 | 82.9 | 81.7 |
| BiLSTM-Mean | 4096 | 79.0 | 78.2 | 83.1 | 81.7 |
| Inner-attention | 4096 | 82.3 | 82.5 | 82.1 | 81.0 |
| HConvNet | 4096 | 83.7 | 83.4 | 82.0 | 80.9 |
| BiLSTM-Max | 4096 | **85.0** | **84.5** | **85.2** | **83.7** |

Table 3: **Performance of sentence encoder architectures** on SNLI and (aggregated) transfer tasks. Dimensions of embeddings were selected according to best aggregated scores (see Figure 5).

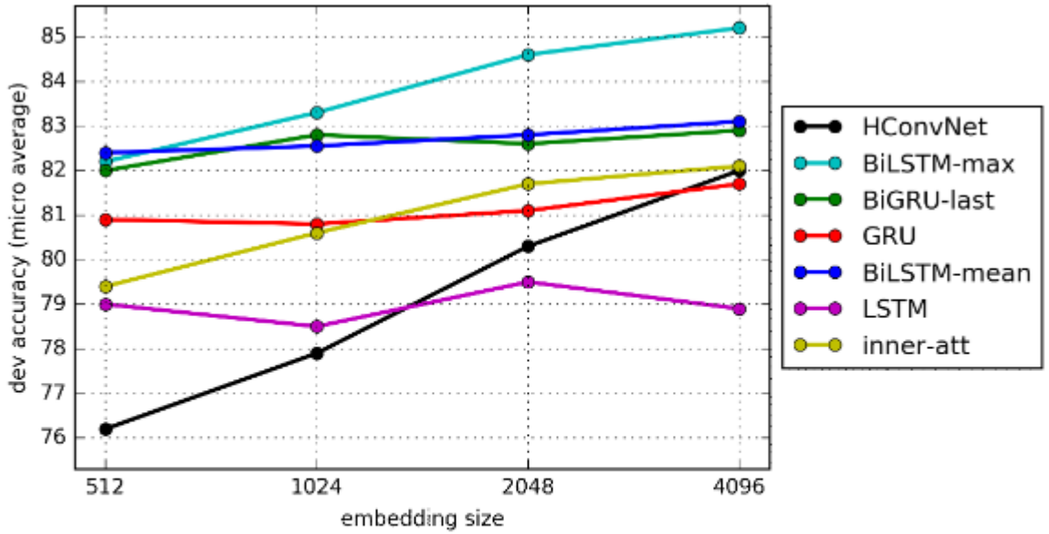



Figure 5: **Transfer performance w.r.t. embedding size** using the micro aggregation method.

# SentenceBERT

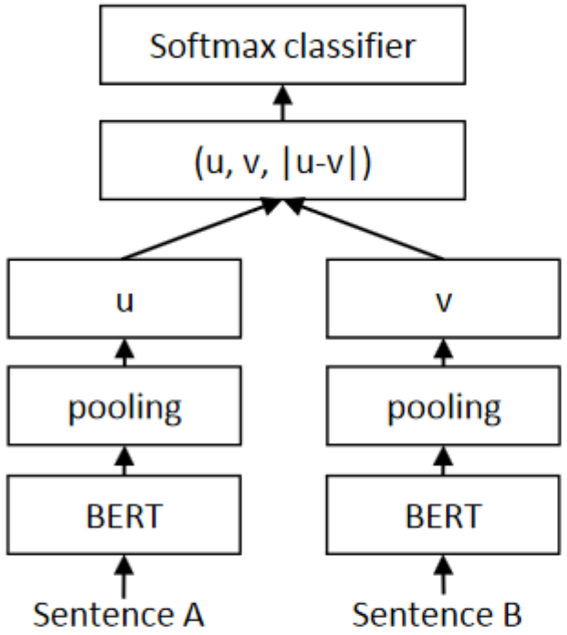



Figure 1: SBERT architecture with classification objective function, e.g., for fine-tuning on SNLI dataset. The two BERT networks have tied weights (siamese network structure).

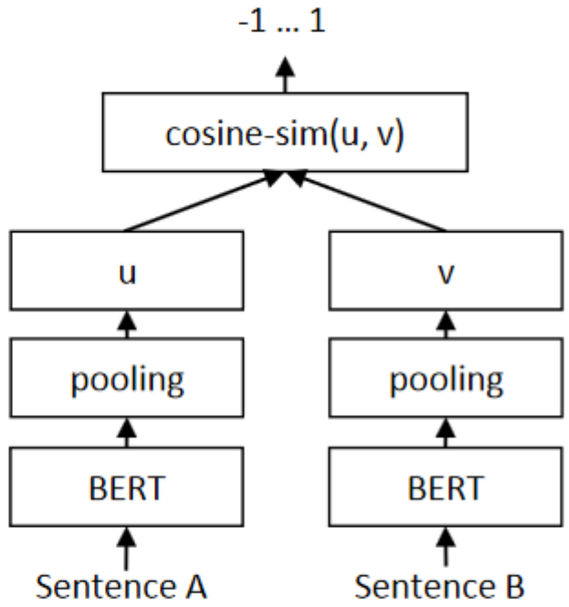



Figure 2: SBERT architecture at inference, for example, to compute similarity scores. This architecture is also used with the regression objective function.

**Nils Reimers, Iryna Gurevych «Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks» // https://arxiv.org/abs/1908.10084**

**Библиотека https://www.sbert.net/**

# SentenceBERT

| Model | STS12 | STS13 | STS14 | STS15 | STS16 | STSb | SICK-R | Avg. |
|---|---|---|---|---|---|---|---|---|
| Avg. GloVe embeddings | 55.14 | 70.66 | 59.73 | 68.25 | 63.66 | 58.02 | 53.76 | 61.32 |
| Avg. BERT embeddings | 38.78 | 57.98 | 57.98 | 63.15 | 61.06 | 46.35 | 58.40 | 54.81 |
| BERT CLS-vector | 20.16 | 30.01 | 20.09 | 36.88 | 38.08 | 16.50 | 42.63 | 29.19 |
| InferSent - Glove | 52.86 | 66.75 | 62.15 | 72.77 | 66.87 | 68.03 | 65.65 | 65.01 |
| Universal Sentence Encoder | 64.49 | 67.80 | 64.61 | 76.83 | 73.18 | 74.92 | **76.69** | 71.22 |
| SBERT-NLI-base | 70.97 | 76.53 | 73.19 | 79.09 | 74.30 | 77.03 | 72.91 | 74.89 |
| SBERT-NLI-large | 72.27 | **78.46** | **74.90** | 80.99 | 76.25 | **79.23** | 73.75 | 76.55 |
| SRoBERTa-NLI-base | 71.54 | 72.49 | 70.80 | 78.74 | 73.69 | 77.77 | 74.46 | 74.21 |
| SRoBERTa-NLI-large | **74.53** | 77.00 | 73.18 | **81.85** | **76.82** | 79.10 | 74.29 | **76.68** |

Table 1: Spearman rank correlation $\rho$ between the cosine similarity of sentence representations and the gold labels for various Textual Similarity (STS) tasks. Performance is reported by convention as $\rho \times 100$. STS12-STS16: SemEval 2012-2016, STSb: STSbenchmark, SICK-R: SICK relatedness dataset.

**Используется BERT / RoBERTa**

**Разные стратегии агрегации: CLS, mean, max**

**Суть: пулинг и обучение сиамской сети**

**Файнтьюним на Stanford Natural Language Inference (SNLI) и Multi-Genre NLI (MNLI)**

**↑ Результат обучения такой архитектуры на разных задачах (проверка на новых задачах)**

# SentenceBERT

| Model | Spearman |
|---|---|
| *Not trained for STS* | |
| Avg. GloVe embeddings | 58.02 |
| Avg. BERT embeddings | 46.35 |
| InferSent - GloVe | 68.03 |
| Universal Sentence Encoder | 74.92 |
| SBERT-NLI-base | 77.03 |
| SBERT-NLI-large | 79.23 |
| *Trained on STS benchmark dataset* | |
| BERT-STSb-base | 84.30 ± 0.76 |
| SBERT-STSb-base | 84.67 ± 0.19 |
| SRoBERTa-STSb-base | **84.92** ± 0.34 |
| BERT-STSb-large | **85.64** ± 0.81 |
| SBERT-STSb-large | 84.45 ± 0.43 |
| SRoBERTa-STSb-large | 85.02 ± 0.76 |
| *Trained on NLI data + STS benchmark data* | |
| BERT-NLI-STSb-base | **88.33** ± 0.19 |
| SBERT-NLI-STSb-base | 85.35 ± 0.17 |
| SRoBERTa-NLI-STSb-base | 84.79 ± 0.38 |
| BERT-NLI-STSb-large | **88.77** ± 0.46 |
| SBERT-NLI-STSb-large | 86.10 ± 0.13 |
| SRoBERTa-NLI-STSb-large | 86.15 ± 0.35 |

Table 2: Evaluation on the STS benchmark test set. BERT systems were trained with 10 random seeds and 4 epochs. SBERT was fine-tuned on the STSb dataset, SBERT-NLI was pretrained on the NLI datasets, then fine-tuned on the STSb dataset.

| | NLI | STSb |
|---|---|---|
| *Pooling Strategy* | | |
| MEAN | **80.78** | **87.44** |
| MAX | 79.07 | 69.92 |
| CLS | 79.80 | 86.62 |
| *Concatenation* | | |
| $(u, v)$ | 66.04 | - |
| $(\|u - v\|)$ | 69.78 | - |
| $(u * v)$ | 70.54 | - |
| $(\|u - v\|, u * v)$ | 78.37 | - |
| $(u, v, u * v)$ | 77.44 | - |
| $(u, v, \|u - v\|)$ | **80.78** | - |
| $(u, v, \|u - v\|, u * v)$ | 80.44 | - |

Table 6: SBERT trained on NLI data with the classification objective function, on the STS benchmark (STSb) with the regression objective function. Configurations are evaluated on the development set of the STSb using cosine-similarity and Spearman's rank correlation. For the concatenation methods, we only report scores with MEAN pooling strategy.

# Топовые обучаемые представления предложений

- **InferSent** (Conneau et al., 2017)
- **Universial Sentence Encoder (USE)** (Cer et al., 2018)
- **SBERT** (Reimers and Gurevych, 2019)

**обучаются на размеченных данных**

# Общий подход и случайный кодировщик

**Вложение предложения ищется в виде** $h = f_\theta(e_1,\ldots,e_n)$

$e_1,\ldots,e_n$ **– вложения слов. Обучаем параметры** $\theta$**.**

| | |
|---|---|
| **InferSent** | $\max(\text{BiLSTM}(e_1,\ldots,e_n))$<br>**Обучаем предсказывая метки «entailment», «neutral», «contradictive»**<br>**cross-entropy** |
| **SkipThought** | $\text{GRU}_n(e_1,\ldots,e_n)$<br>**Декодируем следующее и предыдущее**<br>**negative log-likelihood** |
| **Случайные кодировщики** | |
| **BOPER** | $\text{pool}(We_1,\ldots,We_n)$<br>$W \in \text{rand}([-1/\sqrt{d},+1/\sqrt{d}]\,\vert\,\mathbb{R}^{D\times d})$ |
| **RANDOM LSTM** | $\text{pool}(\text{random\_BiLSTM}(e_1,\ldots,e_n))$ |
| **Echo State Networks (ESNs)** | $\max(\text{ESN}(e_1,\ldots,e_n))$ |

# Случайный кодировщик не сильно хуже!

| Model | Dim | MR | CR | MPQA | SUBJ | SST2 | TREC | SICK-R | SICK-E | MRPC | STSB |
|---|---|---|---|---|---|---|---|---|---|---|---|
| BOE | 300 | 77.3(.2) | 78.6(.3) | 87.6(.1) | 91.3(.1) | 80.0(.5) | 81.5(.8) | 80.2(.1) | 78.7(.1) | 72.9(.3) | 70.5(.1) |
| BOREP | 4096 | 77.4(.4) | 79.5(.2) | 88.3(.2) | 91.9(.2) | 81.8(.4) | **88.8(.3)** | 85.5(.1) | 82.7(.7) | 73.9(.4) | 68.5(.6) |
| RandLSTM | 4096 | 77.2(.3) | 78.7(.5) | 87.9(.1) | 91.9(.2) | 81.5(.3) | 86.5(1.1) | 85.5(.1) | 81.8(.5) | **74.1(.5)** | 72.4(.5) |
| ESN | 4096 | **78.1(.3)** | **80.0(.6)** | **88.5(.2)** | **92.6(.1)** | **83.0(.5)** | 87.9(1.0) | **86.1(.1)** | **83.1(.4)** | 73.4(.4) | **74.4(.3)** |
| InferSent-1 = paper, G | 4096 | 81.1 | 86.3 | 90.2 | 92.4 | 84.6 | 88.2 | 88.3 | 86.3 | 76.2 | 75.6 |
| InferSent-2 = fixed pad, F | 4096 | 79.7 | 84.2 | 89.4 | 92.7 | 84.3 | 90.8 | 88.8 | 86.3 | 76.0 | 78.4 |
| InferSent-3 = fixed pad, G | 4096 | 79.7 | 83.4 | 88.9 | 92.6 | 83.5 | 90.8 | 88.5 | 84.1 | 76.4 | 77.3 |
| Δ InferSent-3, BestRand | - | *1.6* | *3.4* | *0.4* | *0.0* | *0.5* | *2.0* | *2.4* | *1.0* | *2.3* | *2.9* |
| ST-LN | 4800 | 79.4 | 83.1 | 89.3 | 93.7 | 82.9 | 88.4 | 85.8 | 79.5 | 73.2 | 68.9 |
| Δ ST-LN, BestRand | - | *1.3* | *3.1* | *0.8* | *1.1* | *-0.1* | *0.5* | *-0.3* | *-3.6* | *-0.9* | *-5.5* |

Table 1: Performance (accuracy for all tasks except SICK-R and STSB, for which we report Pearson's $r$) on all ten downstream tasks where all models have 4096 dimensions with the exception of BOE (300) and ST-LN (4800). Standard deviations are show in parentheses. InferSent-1 is the paper version with GloVe (G) embeddings, InferSent-2 has fixed padding and uses FastText (F) embeddings, and InferSent-3 has fixed padding and uses GloVe embeddings. We also show the difference between the best random architecture (BestRand) and InferSent-3 and ST-LN, respectively. The average performance difference between the best random architecture and InferSent-3 and ST-LN is 1.7 and -0.4 respectively.

John Wieting, Douwe Kiela No Training Required: Exploring Random Encoders for Sentence Classification
https://arxiv.org/abs/1901.10444

# TSDAE: предтренировка трансформера без меток с шумоподавляющем автокодировщиком

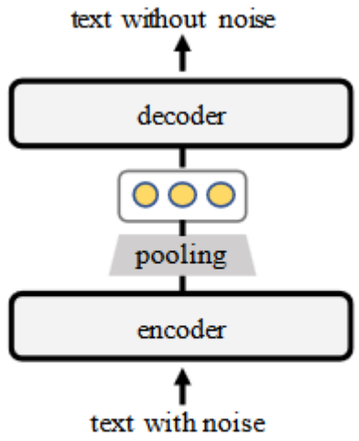



Figure 1: Architecture of TSDAE.

**Архитектура transformer encoder-decoder**

**cross-attention только на фиксированное представление**

**noise (e.g. deleting or swapping words)**

**декодер получает только вектор фиксированной длины**

Kexin Wang et al. «TSDAE: Using Transformer-based Sequential Denoising Auto-Encoder for Unsupervised Sentence Embedding Learning» // https://arxiv.org/pdf/2104.06979v3.pdf

https://github.com/UKPLab/sentence-transformers/

# TSDAE: как зашумляли текст / какой pooling

## лучше удалять с вероятностью 0.6

| Type | Delete | Swap | Mask | Replace | Add |
|---|---|---|---|---|---|
| Score | **78.33** | 76.85 | 76.56 | 74.01 | 72.65 |

Table 5: Results with different noise types

| Ratio | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 |
|---|---|---|---|---|---|---|---|---|---|
| Score | 77.81 | 77.70 | 77.75 | 78.02 | 78.25 | **78.77** | 78.19 | 77.69 | 75.67 |

Table 6: Results with different noise ratio.

## лучше брать представление mean-пулинг

| Method | CLS | Mean | Max |
|---|---|---|---|
| Score | 78.77 | **78.84** | 78.17 |

Table 7: Results with different pooling methods.

# TSDAE: обучение на SNLI + MultiNLI без меток, потом проверка на STS

| Method | STSb | Specific Tasks |
|---|---|---|
| *Unsupervised method* | | |
| TSDAE | 66.0 | **55.2** |
| MLM | 47.3 | 52.9 |
| CT | 73.9 | 52.4 |
| SimCSE | 73.8 | 50.6 |
| BERT-flow | 48.9 | 49.2 |
| *Out-of-the-box supervised pre-trained models* | | |
| SBERT-base-nli-v2 | 83.9 | 52.0 |
| SBERT-base-nli-stsb-v2 | **87.3** | 52.3 |
| USE-large | 80.9 | 54.7 |

Table 3: Performance (Spearman's rank correlation) on the STS benchmark test set. Specific tasks: Average performance from Table 2.

**Качество на STS (semantic textual similarity) не коррелирует с качеством на других задачах**

**Большинство экспериментов на STS (semantic textual similarity) но**

**1) здесь не нужно знание предметной области (domain knowledge)**

новости и заголовки картинок

**2) похожих и непохожих пар примерно поровну**

на практике не так

**3) на практике важно найти именно несколько похожих**

тут другая постановка

| Method / Sub-task/-dataset | AskU. | CQADup. | TwitterP. | | | SciDocs | | | | | Avg. |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | TURL | PIT | Avg. | Cite | CC | CR | CV | Avg. | |
| *Unsupervised learning based on BERT-base* | | | | | | | | | | | |
| TSDAE | 59.4† | 14.5† | 76.8† | 69.2 | 73.0 | 71.4† | 73.9† | 75.0† | 75.6† | 74.0† | 55.2† |
| MLM | 54.3 | 11.7 | 71.9 | 69.7 | 70.8 | 71.2 | 75.8 | 75.1 | 76.2 | 74.6 | 52.9 |
| CT | 56.3 | 13.3 | 74.6 | 70.4 | 72.5 | 63.4 | 67.1 | 70.1 | 69.7 | 67.6 | 52.4 |
| SimCSE | 55.9 | 12.4 | 74.5 | 62.5 | 68.5 | 62.5 | 65.1 | 67.7 | 67.6 | 65.7 | 50.6 |
| BERT-flow | 53.7 | 9.2 | 72.8 | 65.7 | 69.3 | 61.3 | 62.8 | 66.7 | 67.1 | 64.5 | 49.2 |
| *Domain adaptation: NLI+STS → target task* | | | | | | | | | | | |
| TSDAE | 58.7 | 13.6 | 75.8 | 66.2 | 71 | 69.9† | 73.8† | 75† | 75.7† | 73.6† | 54.2† |
| MLM | 54.4 | 9.7 | 69.8 | 68.1 | 69 | 67.1 | 71.8 | 72.6 | 72.9 | 71.1 | 51.1 |
| CT | 57.9 | 14.2 | 75.6 | 70.6 | 73.1 | 62.3 | 66.2 | 68.5 | 68.9 | 66.5 | 52.9 |
| SimCSE | 56.6 | 13.8 | 73.4 | 65.9 | 69.7 | 61.8 | 63.7 | 67.01 | 66.7 | 64.8 | 51.2 |
| BERT-flow | 58.2 | 13.9 | 76.5 | 67.4 | 72 | 62.2 | 64.8 | 68.1 | 68 | 65.8 | 52.5 |
| *Domain adaptation: target task → NLI+STS* | | | | | | | | | | | |
| TSDAE | 59.4† | 14.4† | 75.8 | **73.1**† | **74.5**† | **75.6**† | **78.6**† | **78.1**† | **78.2**† | **77.6**† | **56.5**† |
| MLM | **60.6** | 14.3 | 75.0 | 68.6 | 71.8 | 74.7 | 78.2 | 77.0 | 77.6 | 76.9 | 55.9 |
| CT | 56.4 | 13.4 | 75.9 | 68.9 | 72.4 | 66.5 | 69.6 | 70.6 | 72.2 | 69.7 | 53.0 |
| SimCSE | 56.2 | 13.1 | 75.5 | 67.3 | 71.4 | 65.5 | 68.5 | 70.0 | 71.4 | 68.9 | 52.4 |
| *Other previous unsupervised approaches* | | | | | | | | | | | |
| BM25 | 53.4 | 13.3 | 71.9 | 70.5 | 71.2 | 58.9 | 61.3 | 67.3 | 66.9 | 63.6 | 50.4 |
| Avg. GloVe | 51.0 | 10.0 | 70.1 | 52.1 | 61.1 | 58.8 | 60.6 | 64.2 | 65.4 | 62.2 | 46.1 |
| Sent2Vec | 49.0 | 3.2 | 47.5 | 39.9 | 43.7 | 61.6 | 66.0 | 66.1 | 66.7 | 65.1 | 40.2 |
| BERT-base-uncased | 48.5 | 6.5 | 69.1 | 61.7 | 65.4 | 59.4 | 65.1 | 65.4 | 68.6 | 64.6 | 46.3 |
| *Out-of-the-box supervised pre-trained models* | | | | | | | | | | | |
| SBERT-base-nli-v2 | 53.4 | 11.8 | 75.4 | 69.9 | 72.7 | 66.8 | 70.0 | 70.7 | 72.8 | 70.1 | 52.0 |
| SBERT-base-nli-stsb-v2 | 54.5 | 12.9 | 75.9 | 68.5 | 72.2 | 66.2 | 69.2 | 69.9 | 72.3 | 69.4 | 52.3 |
| USE-large | (*59.3*) | (***15.9***) | **77.1** | 69.8 | 73.5 | 67.1 | 69.5 | 71.4 | 72.6 | 70.2 | 54.7 |
| *In-domain supervised training (upper bound)* | | | | | | | | | | | |
| SBERT-supervised | 63.8 | 16.3 | 81.6 | 75.8 | 78.7 | 90.4 | 91.2 | 86.2 | 83.6 | 87.9 | 61.6 |

Table 2: Evaluation using average precision. Results are averaged over 5 random seeds. The best results excluding the upper bound are bold. USE-large was trained with in-domain training data for AskUbuntu and CQADupStack (scores in italic). Our proposed TSDAE significantly outperforms other unsupervised and supervised out-of-the-box approaches.† marks the cases where TSDAE outperforms both CT and SimCSE in all 5 runs.

**разные домены, «→» – сначала обучение одно, потом другое**

**с разметкой на NLI+STS / без разметки на target task**

# TSDAE: если потом дообучать по типу SBERT

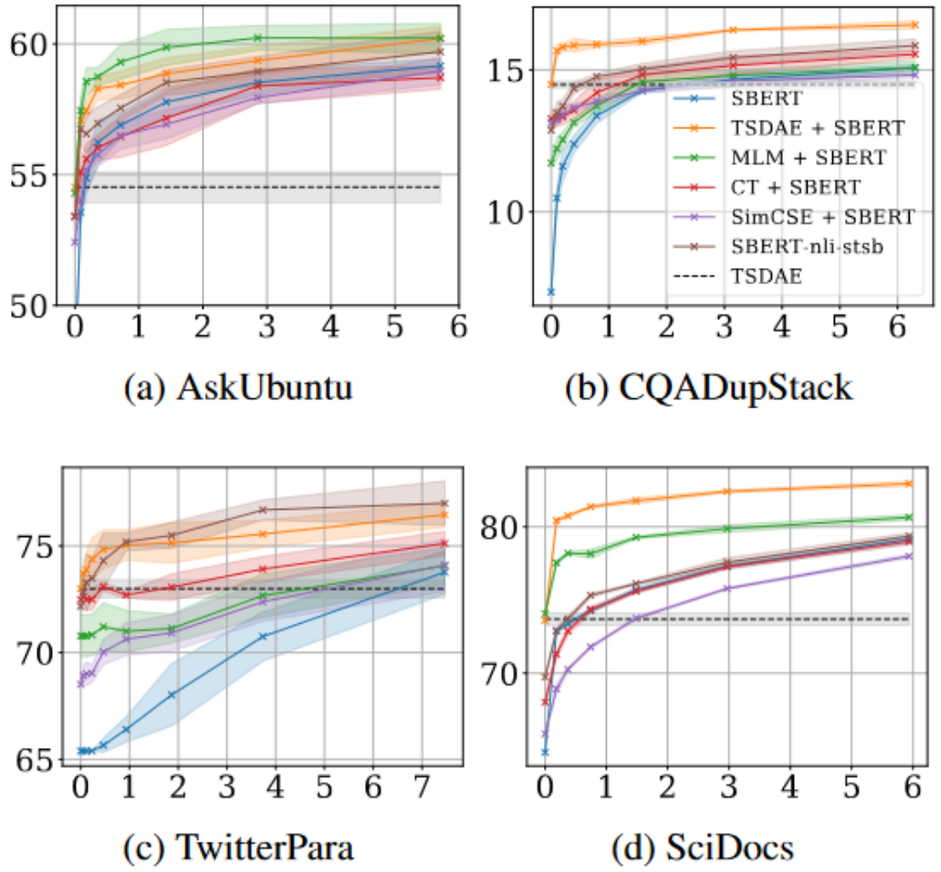



Figure 2: Comparison of different pre-training approaches (TSDAE/MLM/CT/SimCSE+SBERT) with increasing sizes of labeled training data (in thousands). SBERT: Training from the standard *BERT-base-uncased* checkpoint. TSDAE: Unsupervised baseline. Larger plots: Appendix E.

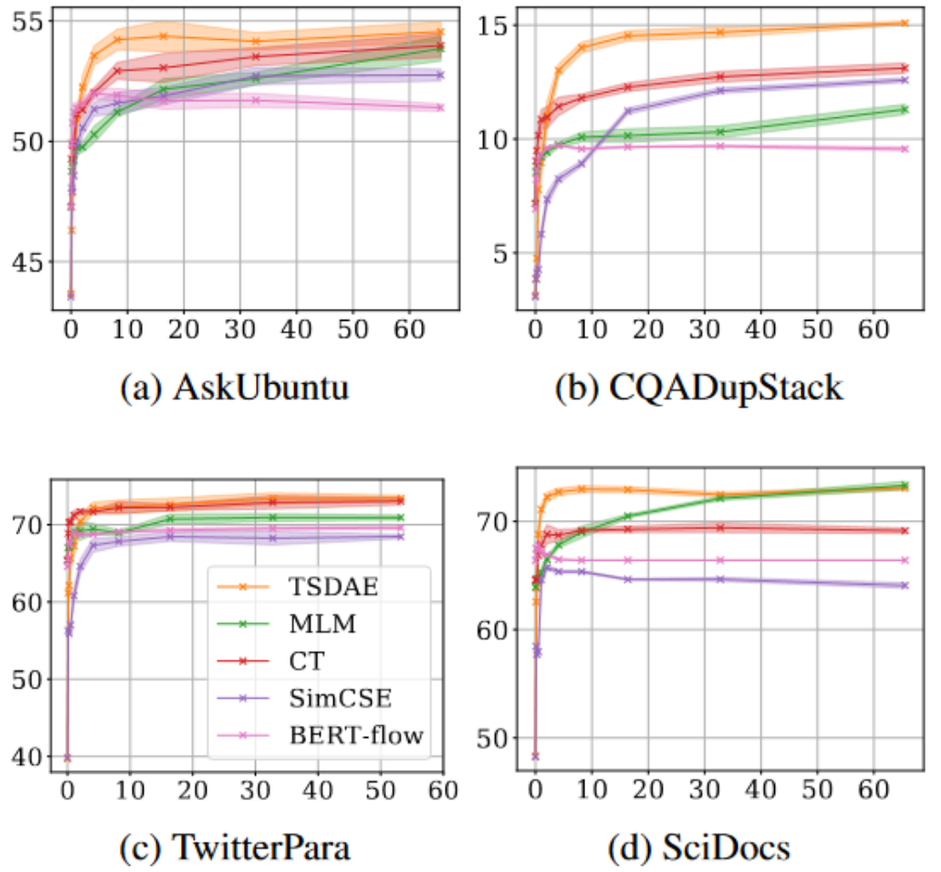



Figure 3: The influence of the number of training sentences (in thousands) on the model performance. Larger plots: Appendix G.

# TSDAE: пробовали разные кодировщики

| Checkpoint | AskU. | CQADup. | TwitterP. | SciDocs | Avg. |
|---|---|---|---|---|---|
| BERT-base | 59.4/2.2 | 14.5/3.4 | 73.0/2.4 | 74.0/2.8 | 55.2/2.7 |
| Scratch | 56.6/2.6 | 8.4/4.2 | 69.8/3.3 | 67.2/3.5 | 50.5/3.4 |
| BART-base | 58.5/1.4 | 9.5/2.0 | 60.3/1.5 | 62.0/1.7 | 47.6/1.7 |
| T5-base | 45.6/1.0 | 2.2/1.4 | 48.2/1.5 | 30.8/1.1 | 31.7/1.3 |

Table 4: Test performance/training loss of TSDAE models starting from different checkpoints. The results for BERT-base are copied from Table 2.

**лучше всего – BERT**

# BERTScore – оценка схожести предложений (чуть-чуть не в тему)

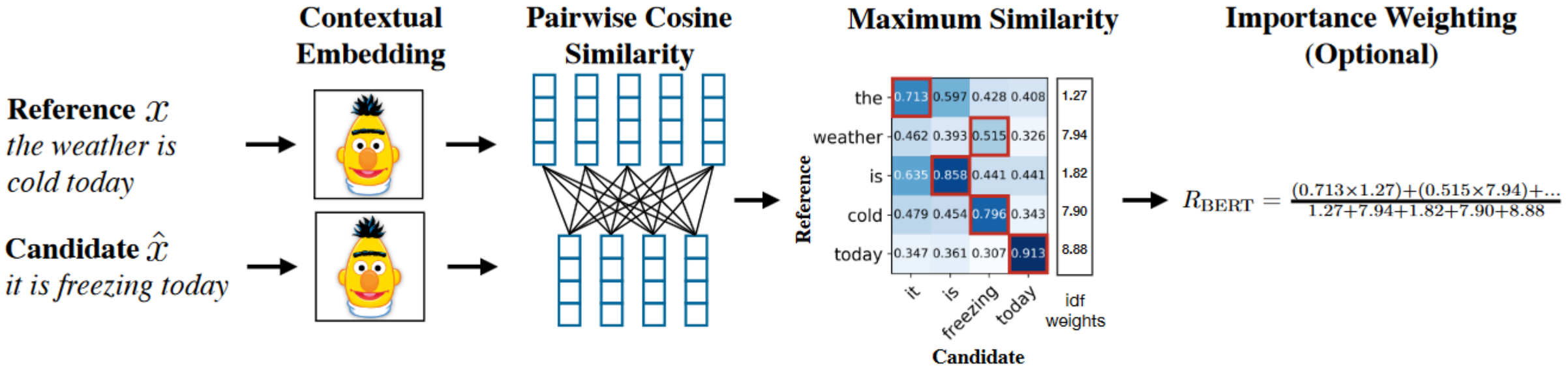



Figure 1: Illustration of the computation of the recall metric $R_{\text{BERT}}$. Given the reference $x$ and candidate $\hat{x}$, we compute BERT embeddings and pairwise cosine similarity. We highlight the greedy matching in red, and include the optional idf importance weighting.

**Tianyi Zhang, et al. «BERTScore: Evaluating Text Generation with BERT»**
**https://arxiv.org/abs/1904.09675**

# Сравнение представлений текстов

Table 6: Results from downstream classification tasks results using a MLP. Values in this table are accuracies for the test set.

| Approach | CR | MPQA | MR | MRPC | SICK-E | SST-2 | SST-5 | SUBJ | TREC |
|---|---|---|---|---|---|---|---|---|---|
| *Baseline* | | | | | | | | | |
| Random Embedding | 61.16 | 68.41 | 48.75 | 64.35 | 54.94 | 49.92 | 24.48 | 49.83 | 18.00 |
| *Experiments* | | | | | | | | | |
| ELMo (BoW, all layers, 5.5B) | 83.95 | **91.02** | **80.91** | 72.93 | 82.36 | **86.71** | 47.60 | **94.69** | 93.60 |
| ELMo (BoW, all layers, original) | 85.11 | 89.55 | 79.72 | 71.65 | 81.86 | 86.33 | **48.73** | 94.32 | 93.40 |
| ELMo (BoW, top layer, original) | 84.13 | 89.30 | 79.36 | 70.20 | 79.64 | 85.28 | 47.33 | 94.06 | 93.40 |
| Word2Vec (BoW, google news) | 79.23 | 88.24 | 77.44 | 73.28 | 79.09 | 80.83 | 44.25 | 90.98 | 83.60 |
| *p*-mean (monolingual) | 80.82 | 89.09 | 78.34 | 73.22 | 83.52 | 84.07 | 44.89 | 92.63 | 88.40 |
| FastText (BoW, common crawl) | 79.63 | 87.99 | 78.03 | 74.49 | 79.28 | 83.31 | 44.34 | 92.19 | 86.20 |
| GloVe (BoW, common crawl) | 78.67 | 87.90 | 77.63 | 73.10 | 79.01 | 81.55 | 45.16 | 91.48 | 84.00 |
| USE (DAN) | 80.50 | 83.53 | 74.03 | 71.77 | 80.39 | 80.34 | 42.17 | 91.93 | 89.60 |
| USE (Transformer) | **86.04** | 86.99 | 80.20 | 72.29 | 83.32 | 86.05 | 48.10 | 93.74 | **93.80** |
| InferSent (AllNLI) | 83.58 | 89.02 | 80.02 | **74.55** | **86.44** | 83.91 | 47.74 | 92.41 | 89.80 |
| SkipThought | 81.03 | 87.06 | 76.60 | 73.22 | 84.33 | 81.77 | 44.80 | 93.33 | 91.00 |

**на разных задачах разные представления успешны**

**ссылка?**

Table 7: Results of the semantic relatedness and textual similarity tasks. Values in this table are the Pearson correlation coefficient for the test sets.

| Approach | SICK-R | STS-12 | STS-13 | STS-14 | STS-15 | STS-16 | STSBenchmark |
|---|---|---|---|---|---|---|---|
| *Experiments* | | | | | | | |
| ELMo (BoW, all layers, 5.5B) | 0.84 | 0.55 | 0.53 | 0.63 | 0.68 | 0.60 | 0.67 |
| ELMo (BoW, all layers, original) | 0.84 | 0.55 | 0.51 | 0.63 | 0.69 | 0.64 | 0.65 |
| ELMo (BoW, top layer, original) | 0.81 | 0.54 | 0.49 | 0.62 | 0.67 | 0.63 | 0.62 |
| Word2Vec (BoW, google news) | 0.80 | 0.52 | 0.58 | 0.66 | 0.68 | 0.65 | 0.64 |
| $p$-mean (monolingual) | 0.86 | 0.54 | 0.52 | 0.63 | 0.66 | 0.67 | 0.72 |
| FastText (BoW, common crawl) | 0.82 | 0.58 | 0.58 | 0.65 | 0.68 | 0.64 | 0.70 |
| GloVe (BoW, common crawl) | 0.80 | 0.52 | 0.50 | 0.55 | 0.56 | 0.51 | 0.65 |
| USE (DAN) | 0.84 | 0.59 | 0.59 | 0.68 | 0.72 | 0.70 | 0.76 |
| USE (Transformer) | 0.86 | **0.61** | **0.64** | **0.71** | **0.74** | **0.74** | **0.78** |
| InferSent (AllNLI) | **0.89** | **0.61** | 0.56 | 0.68 | 0.71 | 0.71 | 0.77 |
| Skip-Thought | 0.86 | 0.41 | 0.29 | 0.40 | 0.46 | 0.52 | 0.75 |

Table 3: Downstream semantic relatedness and textual similarity tasks descriptions and samples.

| Dataset | Task | Sentence 1 | Sentence 2 | Output |
|---|---|---|---|---|
| Microsoft Research Paraphrase Corpus (MRPC) [12] | Classify whether a pair of sentences capture a paraphrase relationship | The procedure is generally performed in the second or third trimester. | The technique is used during the second and, occasionally, third trimester of pregnancy | Paraphase |
| Semantic Text Similarity (STS) [7] | To measure the semantic similarity between two sentences from 0 (not similar) to 5 (very similar) | Liquid ammonia leak kills 15 in Shanghai | Liquid ammonia leak kills at least 15 in Shanghai | 4.6 |

Table 8: Linguistic probing tasks results using a MLP. Values in this table are accuracies on the test set.

| Approach | BShift | CoordInv | ObjNum | SentLen | SOMO | SubjNum | Tense | TopConst | TreeDepth | WC |
|---|---|---|---|---|---|---|---|---|---|---|
| *Baseline* | | | | | | | | | | |
| Random Embedding | 50.16 | 51.38 | 50.82 | 17.07 | 50.44 | 50.79 | 50.02 | 4.71 | 17.57 | 0.12 |
| *Experiments* | | | | | | | | | | |
| ELMo (BoW, all layers, 5.5B) | **85.23** | 69.92 | **89.06** | **89.28** | **59.20** | **91.16** | 89.73 | 84.50 | **48.62** | 88.86 |
| ELMo (BoW, all layers, original) | 84.29 | 69.44 | 88.65 | 89.03 | 58.20 | 90.18 | 90.33 | **84.96** | 48.32 | 89.90 |
| ELMo (BoW, top layer, original) | 81.18 | 68.47 | 87.61 | 78.20 | 58.64 | 90.16 | 88.78 | 81.54 | 44.97 | 72.78 |
| Word2Vec (BoW, google news) | 40.89 | 53.24 | 80.03 | 53.03 | 54.29 | 81.34 | 86.20 | 63.14 | 28.74 | 90.20 |
| $p$-mean (monolingual) | 50.09 | 50.45 | 83.27 | 86.42 | 53.27 | 81.73 | 88.18 | 61.66 | 38.20 | **98.85** |
| FastText (BoW, common crawl) | 50.28 | 53.87 | 80.08 | 66.97 | 55.21 | 80.66 | 87.41 | 67.10 | 36.72 | 91.09 |
| GloVe (BoW, common crawl) | 49.52 | 55.28 | 78.00 | 73.00 | 54.21 | 79.75 | 85.52 | 66.20 | 36.30 | 88.69 |
| USE (DAN) | 60.19 | 54.28 | 69.04 | 57.89 | 55.01 | 71.94 | 80.43 | 60.21 | 25.90 | 60.06 |
| USE (Transformer) | 60.52 | 58.19 | 74.60 | 79.84 | 58.48 | 77.78 | 86.15 | 68.73 | 30.49 | 54.19 |
| InferSent (AllNLI) | 56.64 | 68.34 | 80.54 | 84.13 | 55.79 | 84.45 | 86.74 | 78.34 | 41.02 | 95.18 |
| SkipThought | 70.19 | **71.89** | 83.55 | 86.03 | 54.74 | 86.06 | **90.05** | 82.77 | 41.22 | 79.64 |

Table 4: Linguistic probing tasks description and samples.

| Task | Description | Example | Output |
|---|---|---|---|
| Bigram Shift (BShift) | Whether two words (tokens) in a sentence have been inverted | This is my Eve Christmas . | Inverted |
| Object Number (ObjNum) | Number of the direct object in the main clause (singular and plural) | He received the 200 points . | NNS (Plural) |
| Semantic Odd Man Out (SOMO) | Random noun or verb replaced in the sentence by another noun or verb. Detect whether the sentence has been modified | Tomas surmised as well . | Changed |

Table 9: Results for the image retrieval and caption retrieval tasks using the Microsoft COCO [26] dataset and features extracted with a ResNet-101 [16]. In this table we present Recall at 1 (R@1), Recall at 5 (R@5) and so on, as well as the median.

| | Caption Retrieval | | | | Image Retrieval | | | |
|---|---|---|---|---|---|---|---|---|
| **Approach** | R@1 | R@5 | R@10 | Med r | R@1 | R@5 | R@10 | Med r |
| ELMo (BoW, all layers, 5.5B) | 41.14 | 74.68 | 85.82 | 2.0 | 31.65 | 67.75 | 82.14 | 3.0 |
| ELMo (BoW, all layers, original) | 38.98 | 74.08 | 85.52 | 2.0 | 31.46 | 67.26 | 82.05 | 3.0 |
| ELMo (BoW, top layer, original) | 35.42 | 70.32 | 83.10 | 2.6 | 29.04 | 64.43 | 79.76 | 3.0 |
| Word2Vec (BoW, google news) | 33.82 | 66.56 | 80.32 | 2.8 | 27.18 | 61.91 | 77.77 | 3.8 |
| $p$-mean (monolingual) | 39.18 | 73.40 | 85.22 | 2.0 | 31.34 | 67.11 | 82.02 | 3.0 |
| FastText (BoW, common crawl) | 33.96 | 68.26 | 81.88 | 2.8 | 27.71 | 62.68 | 78.57 | 3.2 |
| GloVe (BoW, common crawl) | 33.96 | 66.08 | 79.42 | 2.8 | 26.70 | 61.18 | 77.35 | 3.8 |
| USE (DAN) | 29.04 | 62.08 | 76.50 | 3.4 | 23.37 | 57.63 | 74.61 | 4.0 |
| USE (Transformer) | 33.48 | 66.74 | 80.42 | 3.0 | 26.96 | 62.34 | 78.33 | 3.4 |
| InferSent (AllNLI) | **42.14** | **75.78** | **87.08** | 2.0 | **33.44** | **69.50** | **83.48** | 3.0 |
| SkipThought | 37.66 | 71.02 | 84.06 | 2.6 | 30.67 | 65.74 | 80.98 | 3.0 |

# Бонус: сексизм в представлениях

**есть классические равенства**

$$\overrightarrow{\text{man}} - \overrightarrow{\text{woman}} \approx \overrightarrow{\text{king}} - \overrightarrow{\text{queen}}$$

**но есть также**

$$\overrightarrow{\text{man}} - \overrightarrow{\text{woman}} \approx \overrightarrow{\text{computer programmer}} - \overrightarrow{\text{homemaker}}$$

| Extreme *she* occupations | | | Extreme *he* occupations | | |
|---|---|---|---|---|---|
| 1. homemaker | 2. nurse | 3. receptionist | 1. maestro | 2. skipper | 3. protege |
| 4. librarian | 5. socialite | 6. hairdresser | 4. philosopher | 5. captain | 6. architect |
| 7. nanny | 8. bookkeeper | 9. stylist | 7. financier | 8. warrior | 9. broadcaster |
| 10. housekeeper | 11. interior designer | 12. guidance counselor | 10. magician | 11. figher pilot | 12. boss |

Figure 1: The most extreme occupations as projected on to the *she*−*he* gender direction on g2vNEWS. Occupations such as *businesswoman*, where gender is suggested by the orthography, were excluded.

**Tolga Bolukbasi et al. «Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings» // https://arxiv.org/pdf/1607.06520.pdf**

# Бонус: сексизм в представлениях

**Gender stereotype *she-he* analogies.**

| | | |
|---|---|---|
| sewing-carpentry | register-nurse-physician | housewife-shopkeeper |
| nurse-surgeon | interior designer-architect | softball-baseball |
| blond-burly | feminism-conservatism | cosmetics-pharmaceuticals |
| giggle-chuckle | vocalist-guitarist | petite-lanky |
| sassy-snappy | diva-superstar | charming-affable |
| volleyball-football | cupcakes-pizzas | hairdresser-barber |

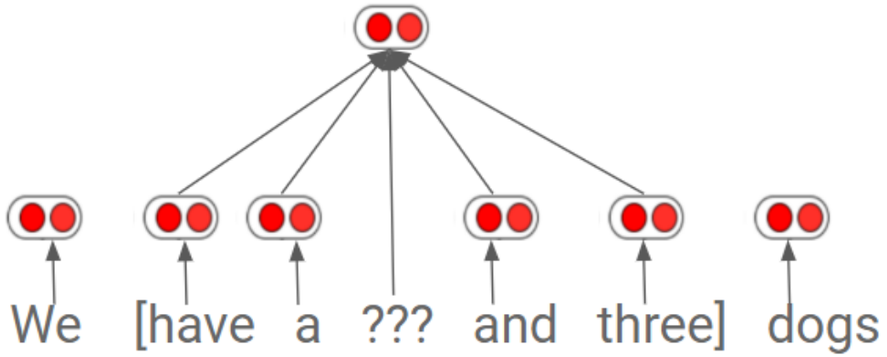
**Gender appropriate *she-he* analogies.**

| | | |
|---|---|---|
| queen-king | sister-brother | mother-father |
| waitress-waiter | ovarian cancer-prostate cancer | convent-monastery |

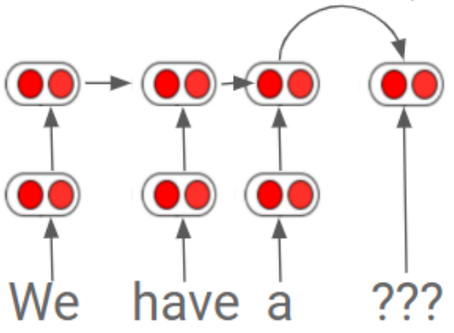
Figure 2: **Analogy examples**. Examples of automatically generated analogies for the pair *she-he* using the procedure described in text. For example, the first analogy is interpreted as *she*:*sewing* :: *he*:*carpentry* in the original w2vNEWS embedding. Each automatically generated analogy is evaluated by 10 crowd-workers are to whether or not it reflects gender stereotype. Top: illustrative gender stereotypic analogies automatically generated from w2vNEWS, as rated by at least 5 of the 10 crowd-workers. Bottom: illustrative generated gender-appropriate analogies.

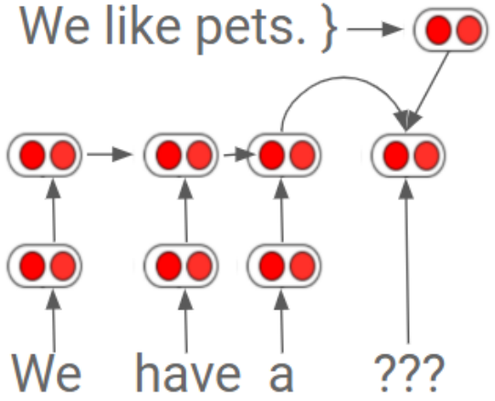
# Итог

word2vec, Mikolov et al (2013)



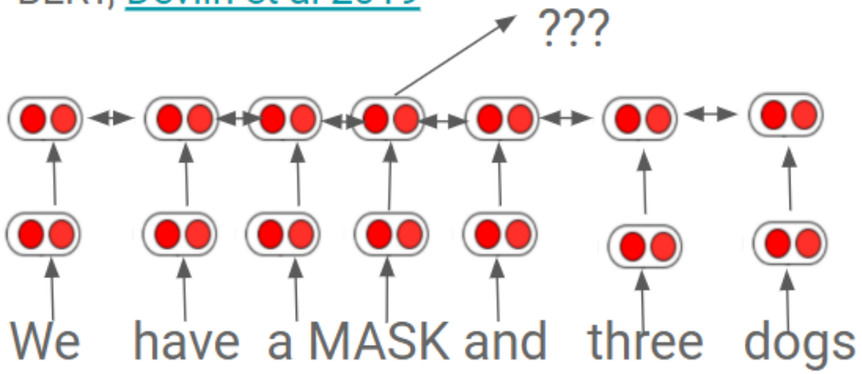

ELMo, Peters et al. 2018, ULMFiT (Howard & Ruder 2018), GPT (Radford et al. 2018)




Skip-Thought (Kiros et al., 2015)

BERT, Devlin et al 2019




**http://tiny.cc/NAACLTransfer**

# Итог

**Есть классические испытанные способы**

**Они используются и для получения более продвинутых представлений**

**Есть способы учёта контекста**

**дальше будем ещё с этим работать**

**Можно получать представления целых предложений / текстов**

**Важно: представления можно обучать вместе с моделью**

# Ссылки

**Поддерживаемый каталог представлений**

https://github.com/Separius/awesome-sentence-embedding

**хорошо тонкости методов расписаны**

https://lilianweng.github.io/lil-log/2017/10/15/learning-word-embedding.html

**хороший обзор по этой теме**

https://lena-voita.github.io/nlp_course/

**Хороший туториал по w2v**

http://mccormickml.com/2017/01/11/word2vec-tutorial-part-2-negative-sampling/

**Коллекция старых представлений**

https://github.com/Separius/awesome-sentence-embedding