# Глубокое обучение
# Обработка текстов

**Дьяконов А.Г.**

**Московский государственный университет имени М.В. Ломоносова (Москва, Россия)**

# Задачи

## текст → метка / метки

Определение темы / настроения / автора

Определение тональности

Разметка на части речи

## Текст → текст

Машинный перевод

Аннотирование

Чат-бот

## текст, текст → текст

ответы на вопросы

справочная / экспертная система

## … → текст

описание изображения

# Способы кодирования слов

- **ONE**

- **counts (сумма ONE соседей)**

**более нетривиальная оценка близости с помощью cos**

- **вложение (embeddings)**

**умный алгоритм задания кодировки**

## word embeddings

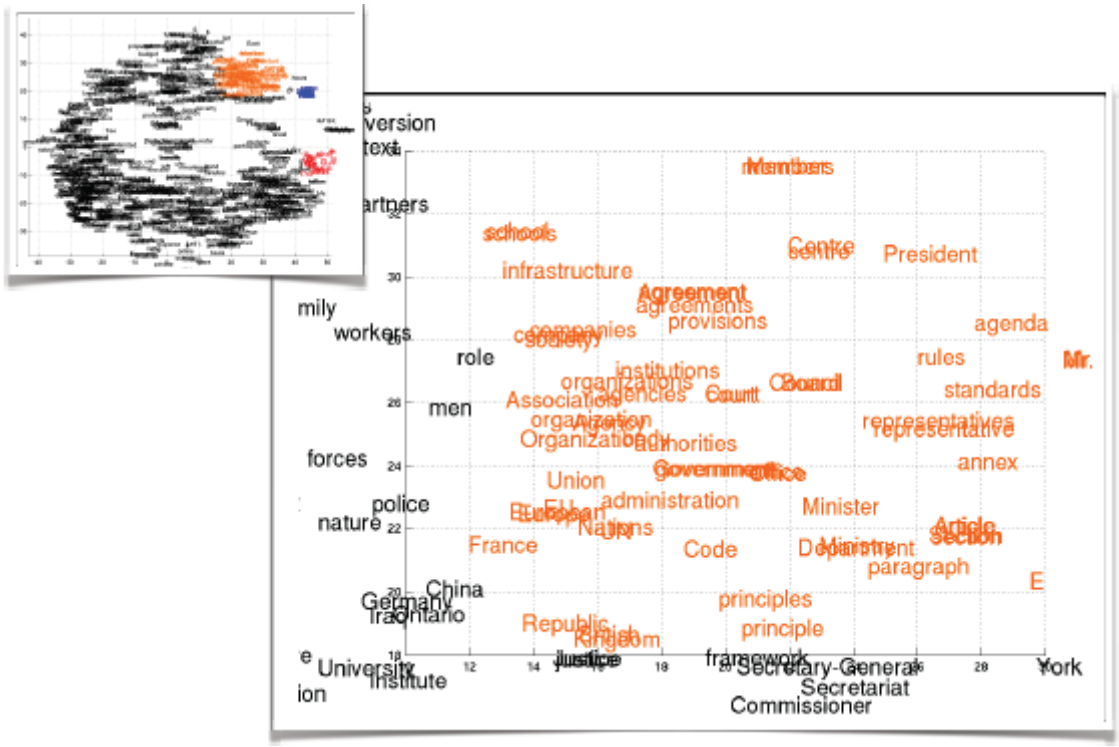
**Представления слов в вещественном многомерном пространстве**

**⇒ можно использовать в матмоделях**

**Предобученные**

**Обученные для конкретной задачи**

# Вложение слов в непрерывное пространство (embedding)




**Несколько популярных способов**

- **word2vec** [Mikolov et al. 2013] предсказания слово ↔ контекст
- **fasttext** = word2vec + ngrams
- **Glove** [Pennington et al. 2014] обучение весов слов через разложение матрицы совместной встречаемости

# word2vec

**Трюк: настраиваем модель, но не для использования в задаче, которой учим (нас интересуют формируемые внутренние представления)** Аналогично было в автокодировщиках;)

**Термины «distributional semantics»**

**Смысл слова определяется контекстом**

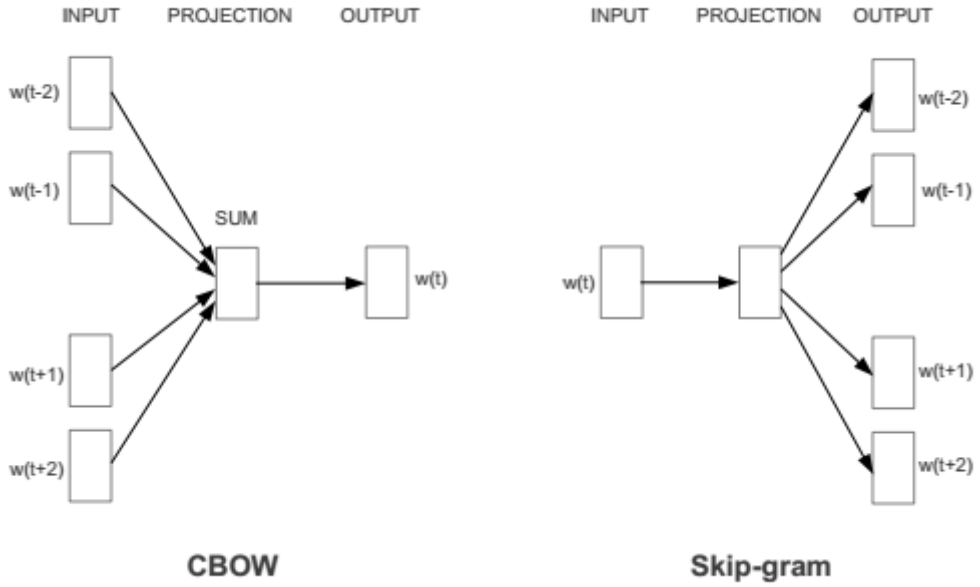
Полосатая маленькая ***** мурлычит и пьёт молоко

Весна

Ручьи
Тает
Цветёт
Зеленеет
Прилетают

# word2vec

## Два представления контекста:




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

## Два метода обучения: позже

- **Hierarchical Softmax**
- **Negative Sampling**

# word2vec

**Предсказываем слово по контексту**

**используется реже, чем следующая реализация**

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

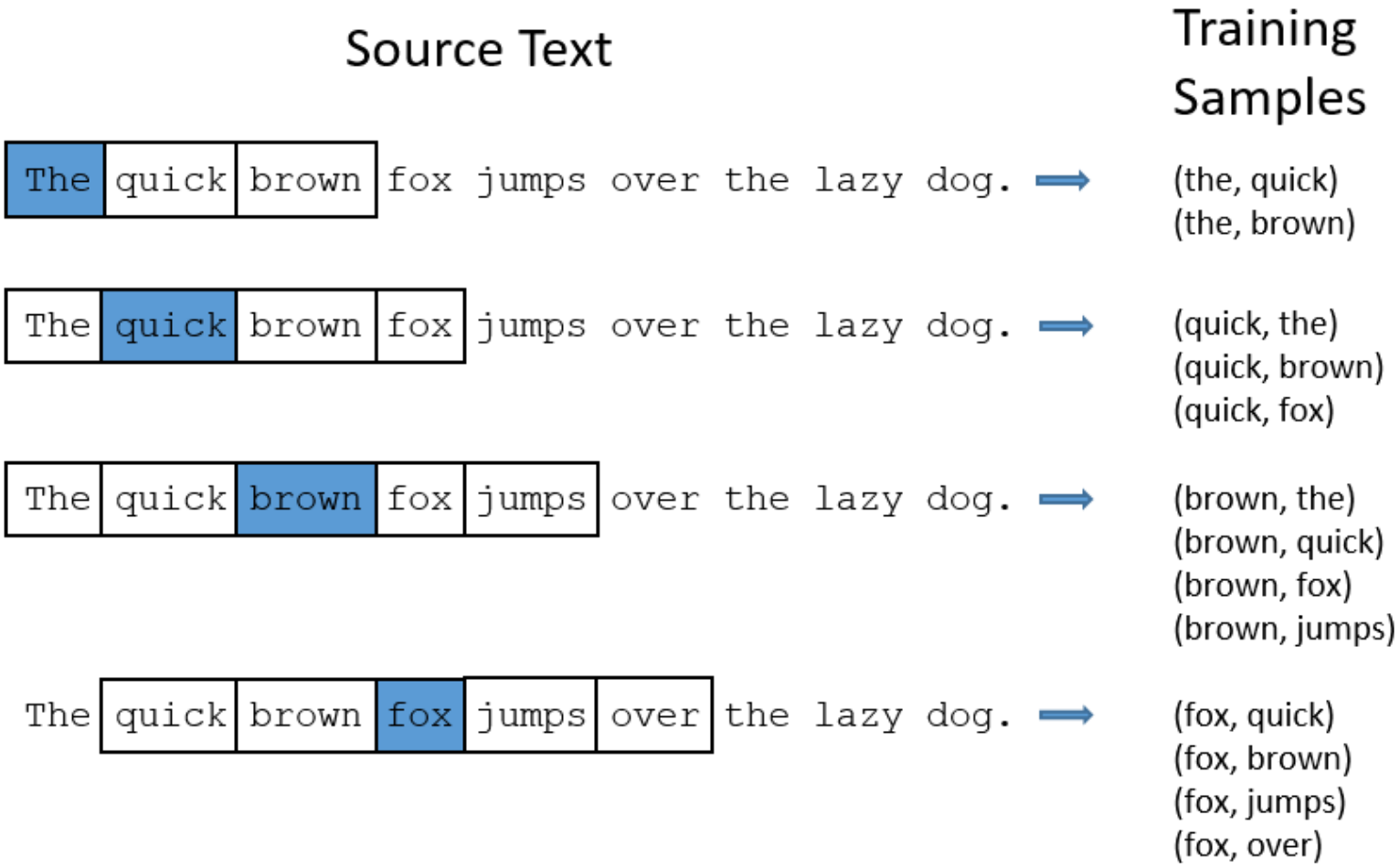
**выделено то, что будем считать кодировкой**

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

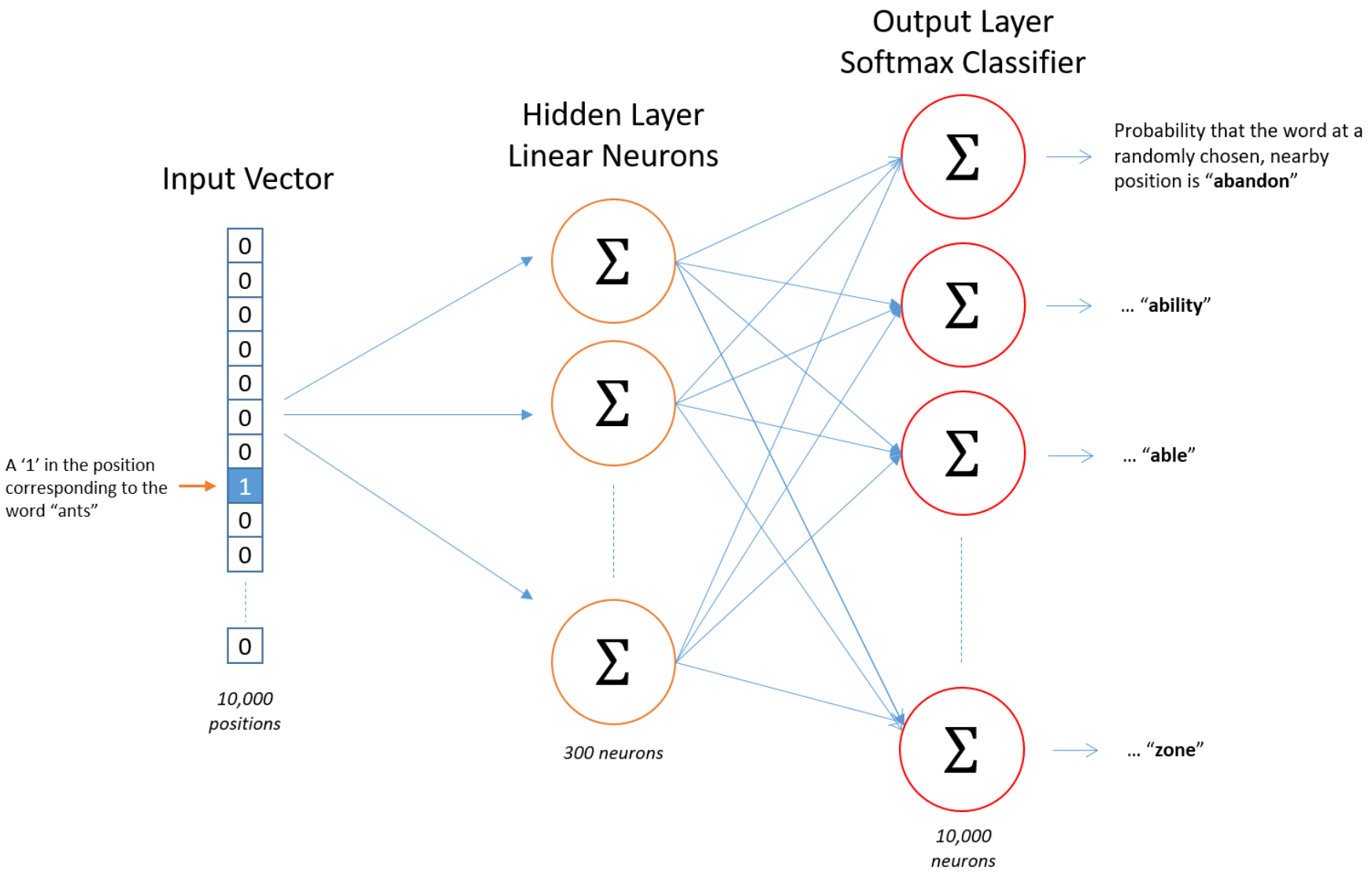
http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/

# word2vec

## Предсказываем контекст по слову



# word2vec




**вход: ONE-кодировка слова выход: распределение вероятностей**

**Средний слой – для нашего кодирования**

# word2vec

**Огромная НС**

**Первый слой – #слов × размерность предствления**

**Как обучать????**

«Distributed Representations of Words and Phrases and their Compositionality» [Mikolov T. 2013 https://arxiv.org/pdf/1310.4546.pdf]

Следующие слайды по

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

**Есть отличия между реализацией и статьёй!**

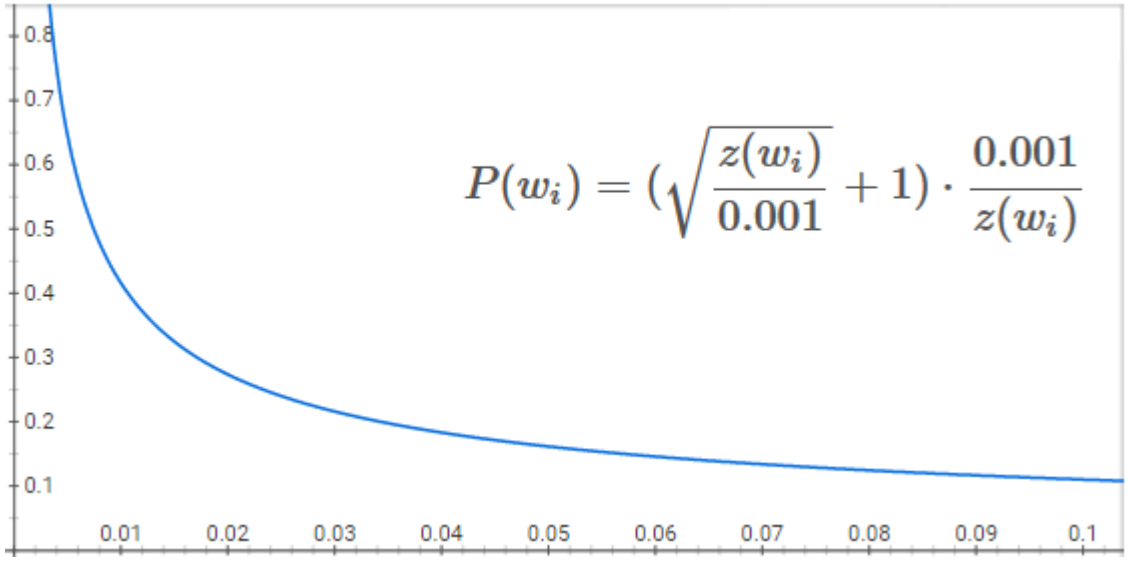
## word2vec

**Распространённые фразы – одно слово**

White_Spunner_Construction
Bad_Habits
Toxics_Alliance

**Частые слова – реже выбираются при обучении**

вероятность быть выбранным от частоты:




**«Negative Sampling»**

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

# word2vec – немного математики

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

**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 – немного математики

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

# Ближайшие соседи

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

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

# Операции над представлениями слов

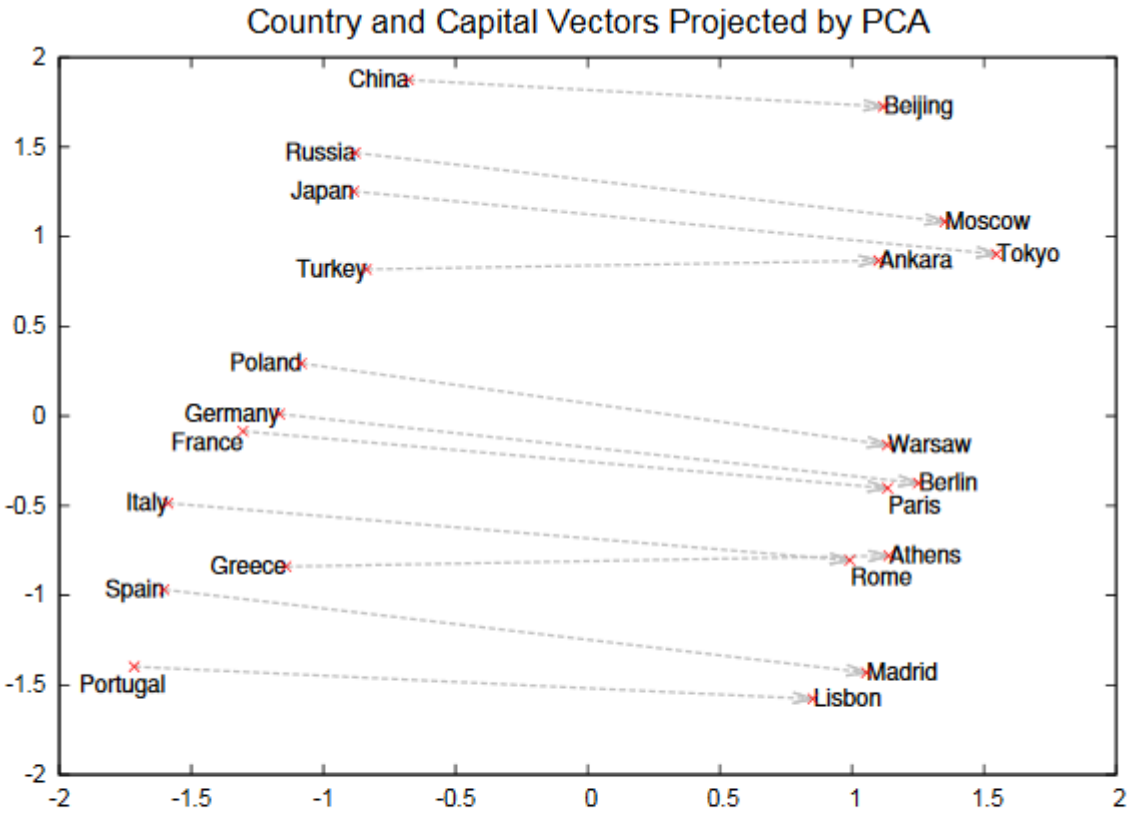



Figure 2: Two-dimensional PCA projection of the 1000-dimensional Skip-gram vectors of countries and their capital cities. The figure illustrates ability of the model to automatically organize concepts and learn implicitly the relationships between them, as during the training we did not provide any supervised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

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

# Другие представления

**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;)**

«Enriching Word Vectors with Subword Information» [Bojanowski P. et al., 2017 https://arxiv.org/pdf/1607.04606.pdf]

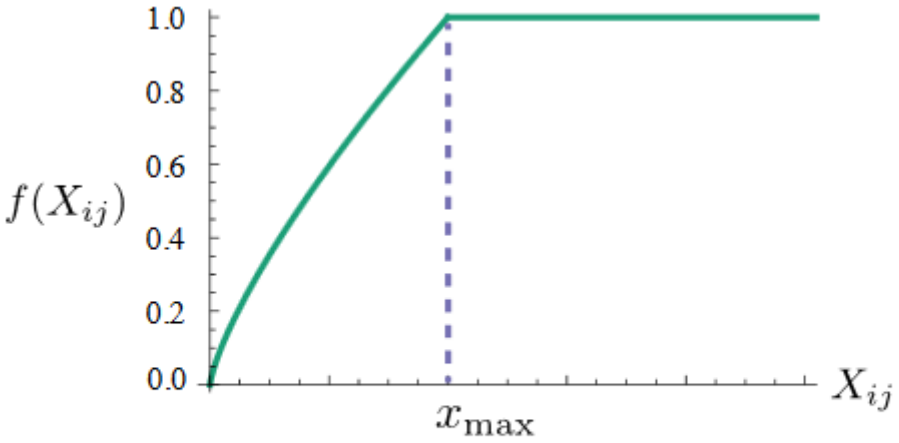
**https://fasttext.cc – тут есть все ссылки!!!**

# Glove: Global Vectors for Word Representation

**Пусть** $\| p_{ij} \|_{m\times m}$ **– матрица встречаемости**

$$h_{ij} = p(j \mid i) = \frac{\#i}{\#ij}$$

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




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

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

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

# Glove: ближайшие соседи

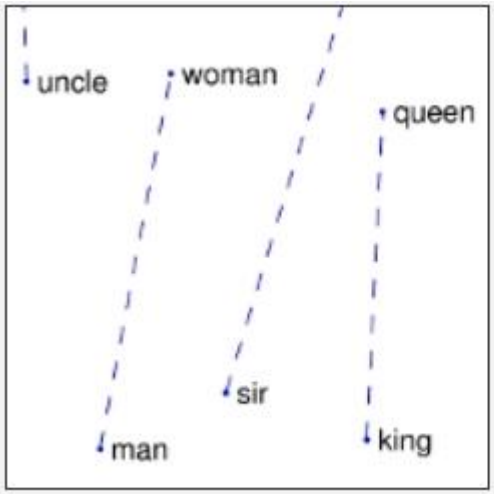
**frog**
**frogs**
**toad**
**litoria**
**leptodactylidae**
**rana**
**lizard**
**leutherodactylus**


3. litoria

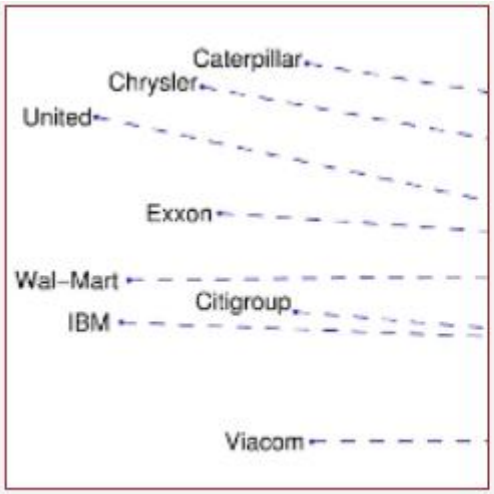

4. leptodactylidae


5. rana

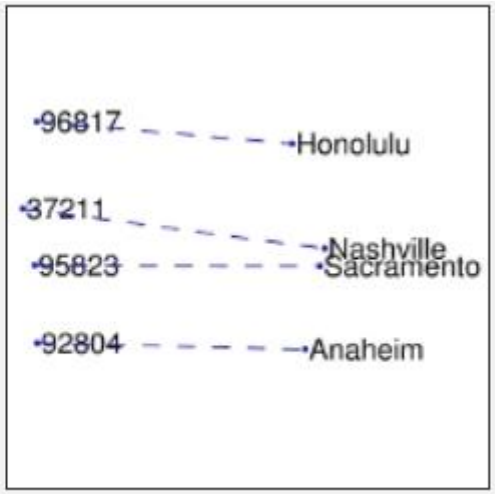

7. eleutherodactylus



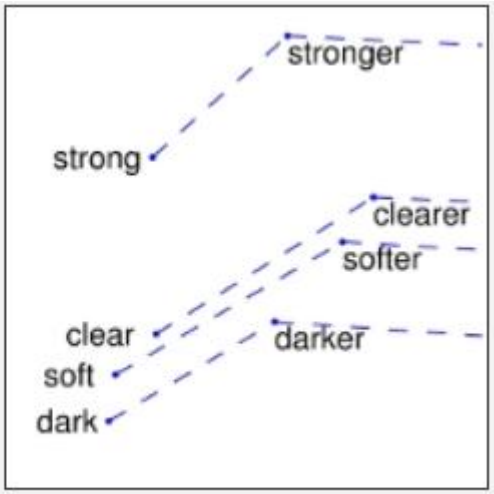
man - woman



company - ceo



city - zip code



comparative - superlative

# Модель seq2seq

**Как переводить последовательность → последовательность**

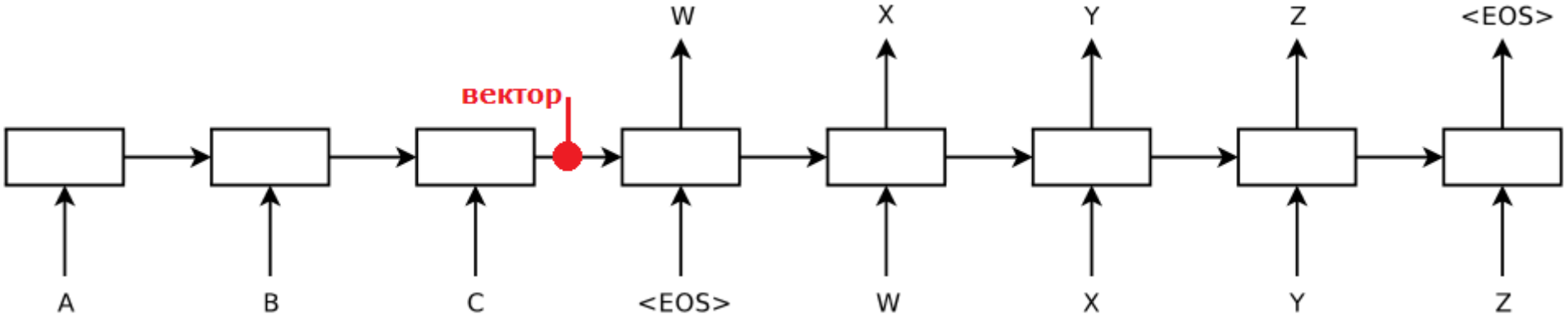
**Многослойная (4 слоя) LSTM**

**последовательность → вектор**

**Другая (так, понятно, лучше!) многослойная LSTM**

**вектор → целевая последовательность**

**Интересно: в задаче перевода качество повышало инвертирование порядка входа!**

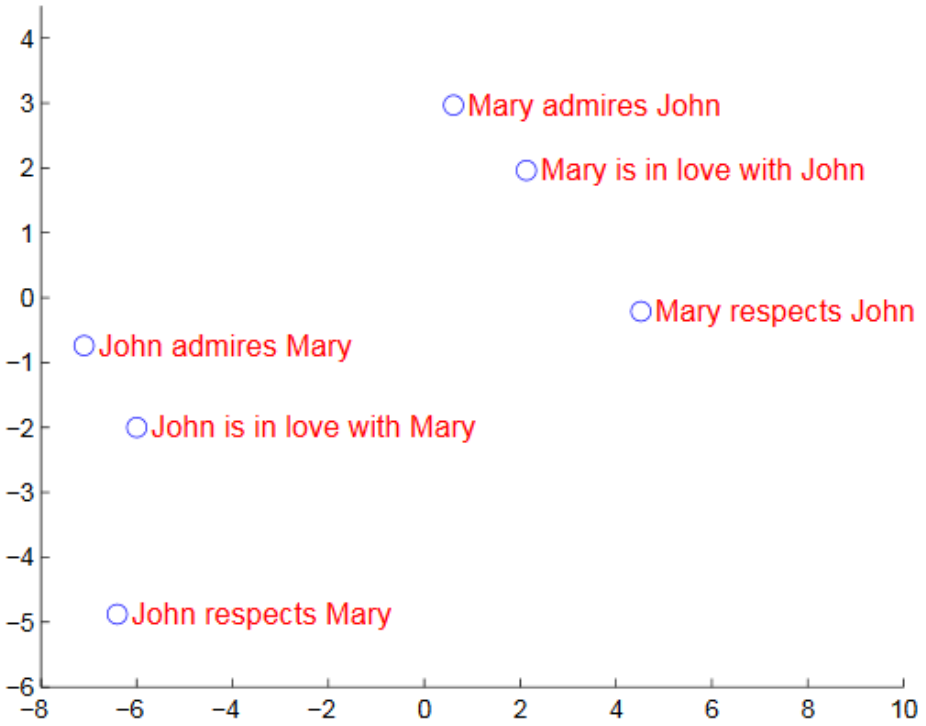


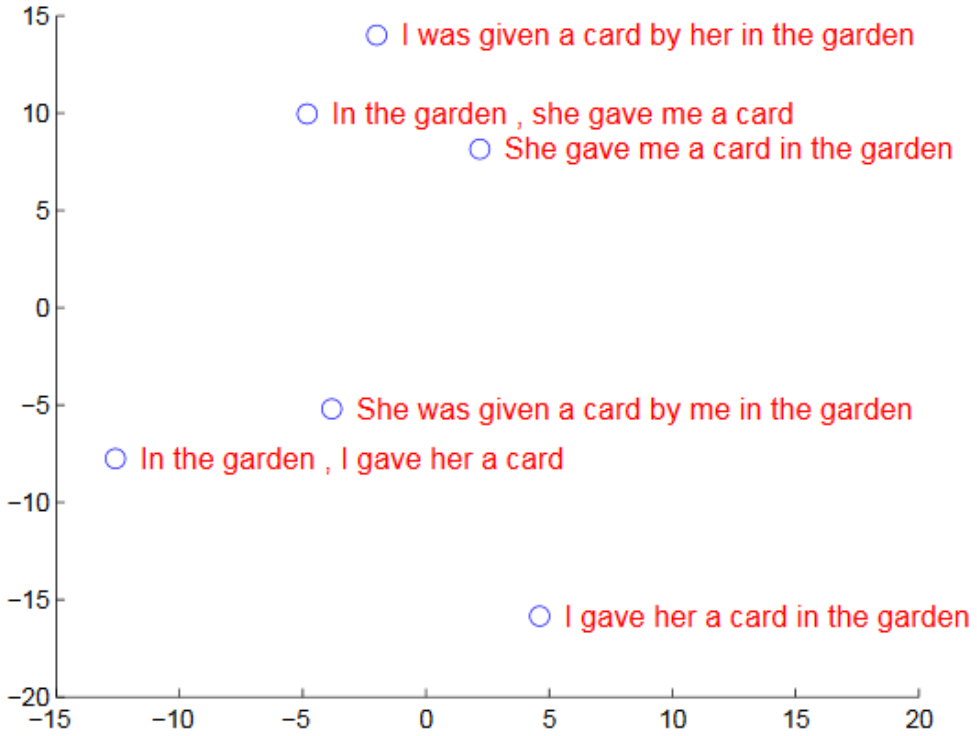

**«Sequence to Sequence Learning with Neural Networks»**

**[Sutskever I. и др. 2014, https://arxiv.org/abs/1409.3215]**

# Модель seq2seq

## Внутреннее представление предложений!




**left-to-right beam-search decode**

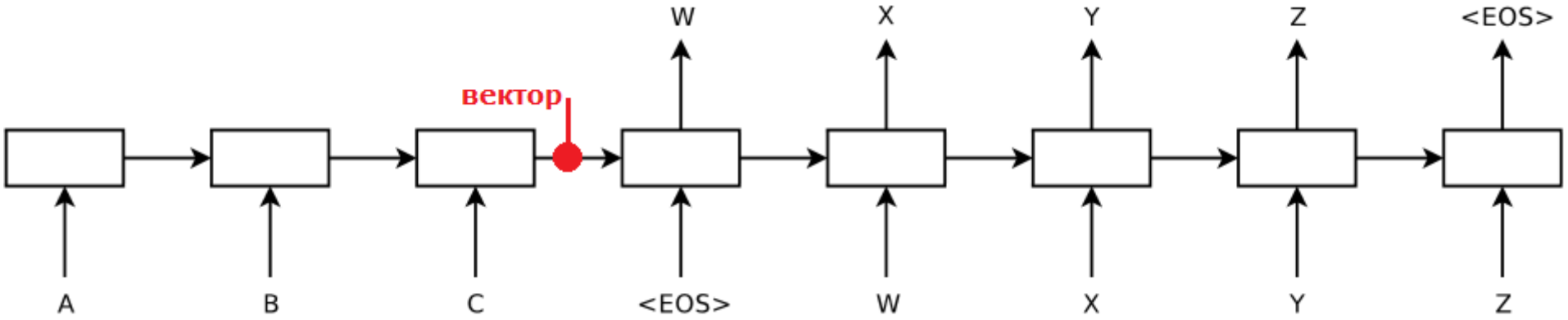
**если выбираем лучшего следующего, не обязательно максимизируем качество**

**Обучение 10 дней**

**Тоже хороши ансамбли**

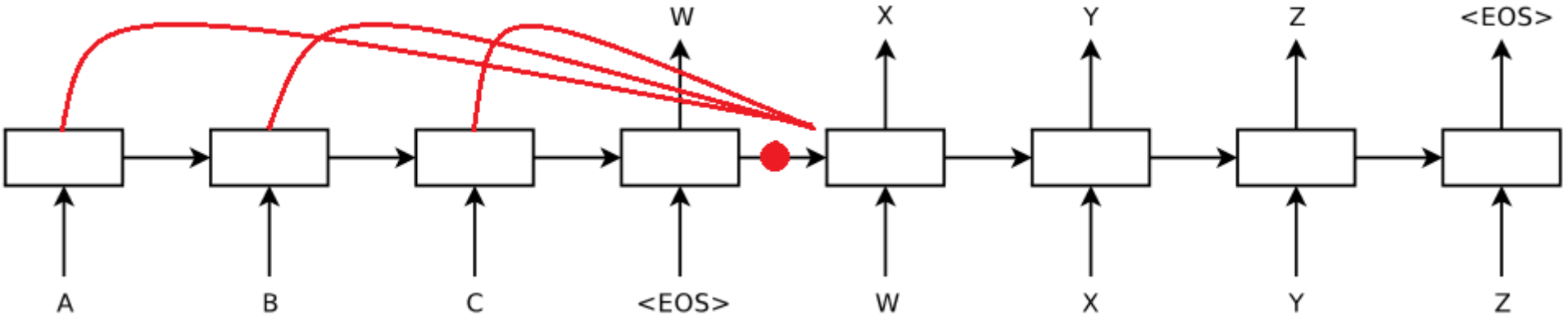
# Обобщения seq2seq

**На одном нейроне вся информация о тексте… плохо**
**(особенно для длинных последовательностей)**

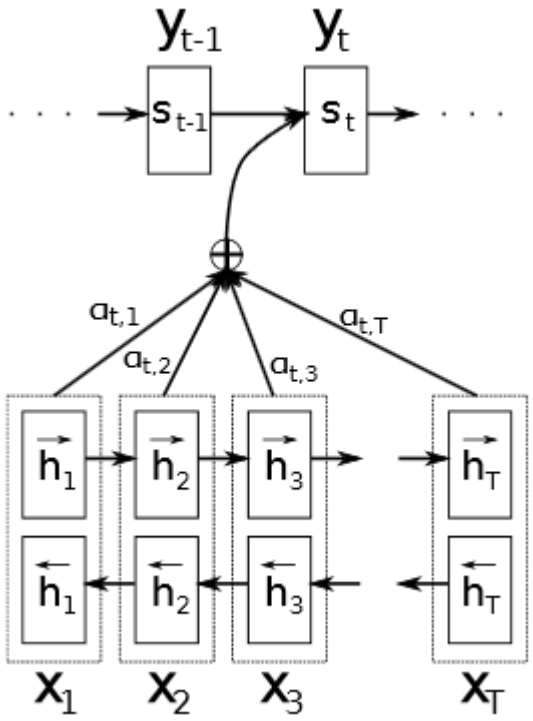



**Решение – механизм внимания**

**частично был в RNN**



# Механизм внимания




**Не будем пытаться закодировать всё предложение одним вектором!**

**Добавляется контекстный вектор**

$$c_i = \sum_j \alpha_{ij} h_j$$

**веса**

$$\alpha_{ij} = \exp(e_{ij}) / \sum_k \exp(e_{ik})$$

**Насколько соответствуют состояния**

$$e_{ij} = a(s_{i-1}, h_j)$$

**Учитываются не только слова ДО, но и ПОСЛЕ!**

**Конкатенация состояния ДО и состояния ПОСЛЕ**

**Bidirectional RNN (BiRNN)**

# Механизм внимания

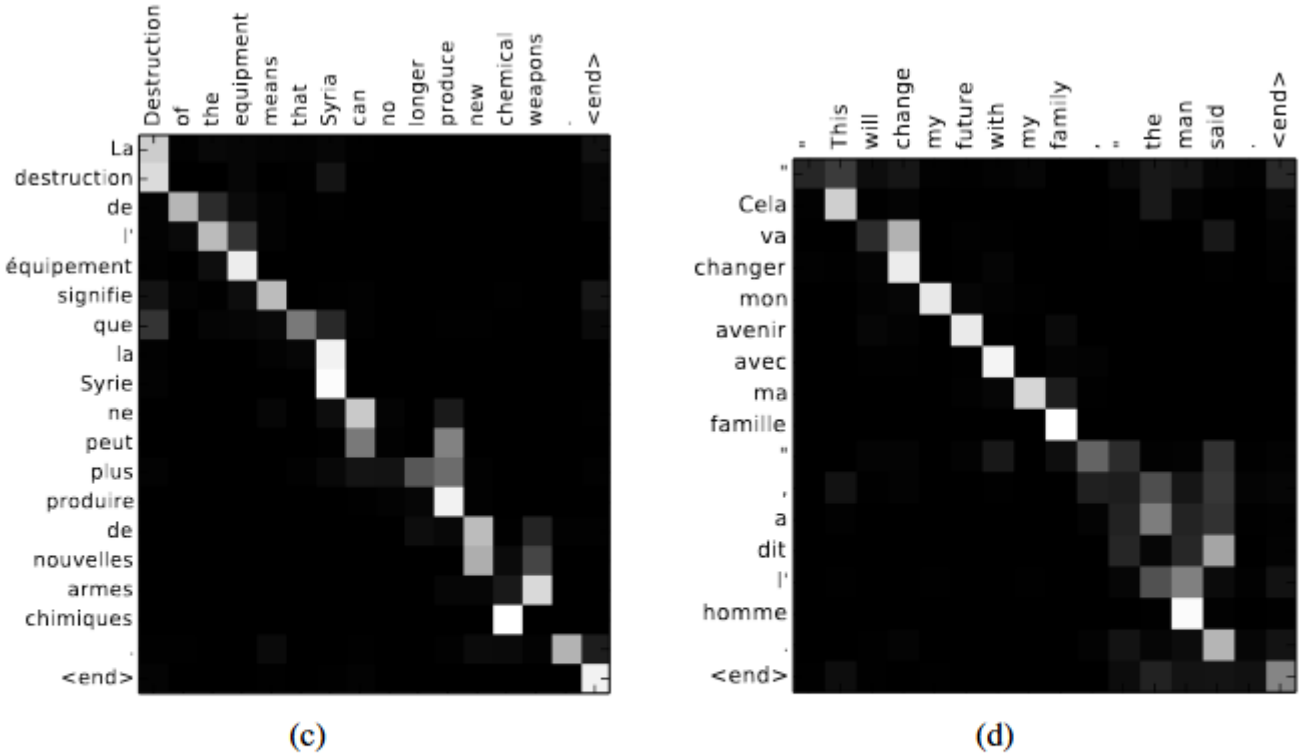



Figure 3: Four sample alignments found by RNNsearch-50. The x-axis and y-axis of each plot correspond to the words in the source sentence (English) and the generated translation (French), respectively. Each pixel shows the weight $\alpha_{ij}$ of the annotation of the $j$-th source word for the $i$-th target word (see Eq. (6)), in grayscale (0: black, 1: white). (a) an arbitrary sentence. (b–d) three randomly selected samples among the sentences without any unknown words and of length between 10 and 20 words from the test set.

**«Neural Machine Translation by Jointly Learning to Align and Translate»**
**[Bahdanau D. и др., 2016 https://arxiv.org/abs/1409.0473]**

# Memory Network «MemN2N» – использование памяти

## Задача

Sam walks into the kitchen.
Sam picks up an apple.
Sam walks into the bedroom.
Sam drops the apple.

**Q: Where is the apple?**

A. Bedroom

Brian is a lion.
Julius is a lion.
Julius is white.
Bernhard is green.

**Q: What color is Brian?**

A. White

Mary journeyed to the den.
Mary went back to the kitchen.
John journeyed to the bedroom.
Mary discarded the milk.

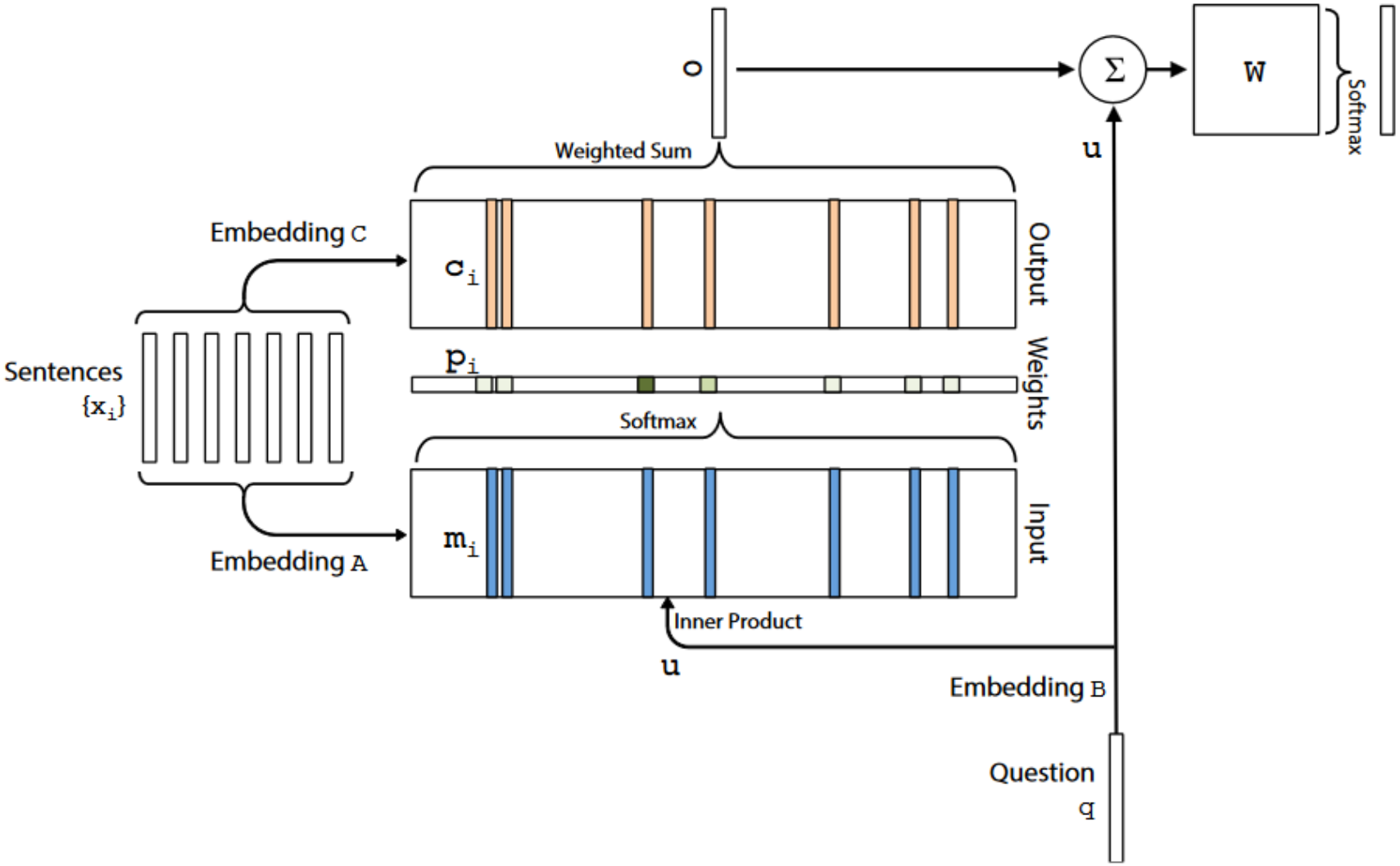
**Q: Where was the milk before the den?**

A. Hallway

https://github.com/facebook/MemNN

# Memory Network «MemN2N» – использование памяти

## Однослойный пример




«End-To-End Memory Networks» [Sukhbaatar S. и др., 2015 https://arxiv.org/abs/1503.08895]

# Memory Network «MemN2N» – использование памяти

**Задача: дан текст** $x_1, \ldots, x_T$
**и вопрос** $q$**. Надо дать ответ** $a$**.**

**Пользуемся вложениями:**

$$x_i \to m_i \in \mathbb{R}^d \quad \textbf{(1)}$$

$$q \to u \in \mathbb{R}^d \quad \textbf{(2)}$$

**Релевантности запроса тексту:**

$$\{u^{\mathrm{T}} m_i\}_i \xrightarrow[\text{softmax}]{} \{p_i\}_i$$

**Есть другое вложение (для подготовки ответа)**

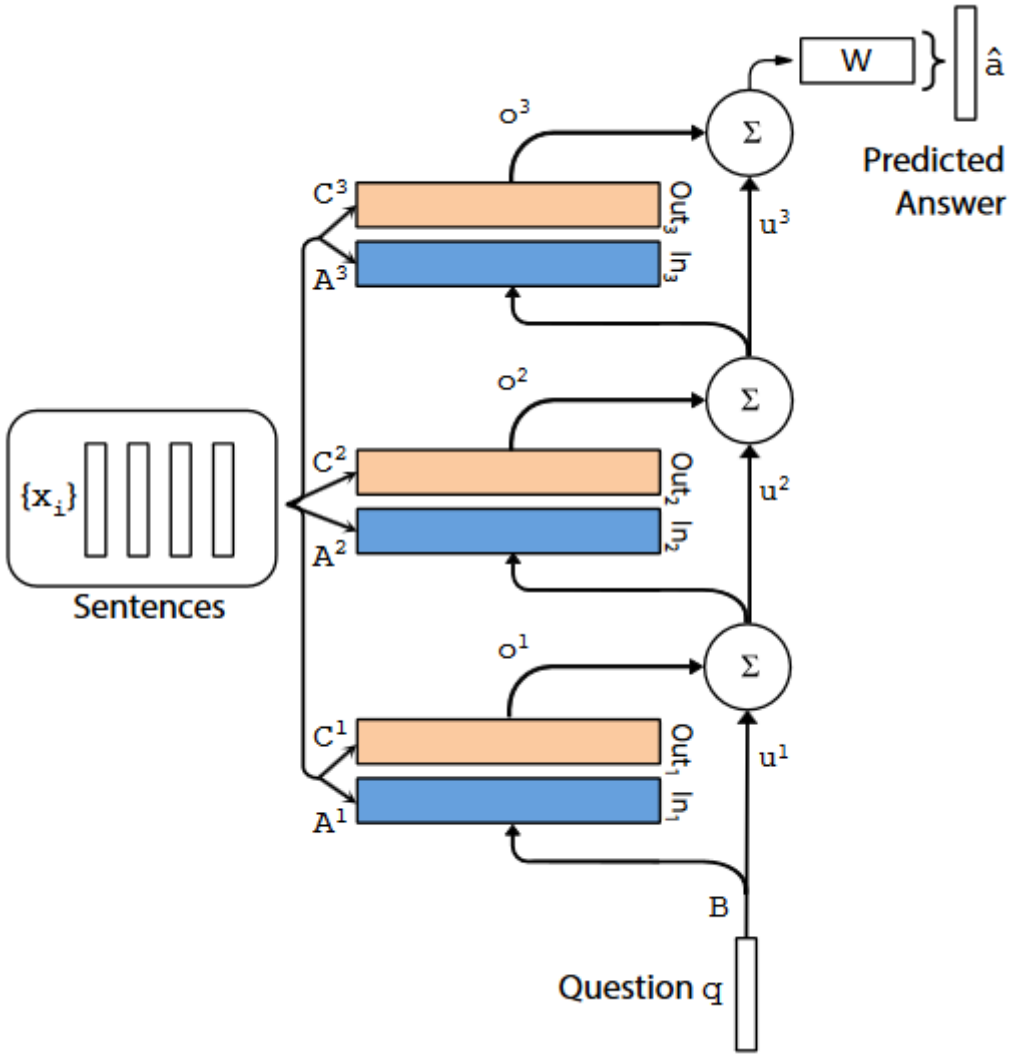
$$x_i \to c_i \in \mathbb{R}^d \quad \textbf{(3)}$$

**Ответ:**

$$\text{softmax}\left(W\left(\sum_i p_i c_i + u\right)\right)$$

# Memory Network «MemN2N» – использование памяти

## Можно использовать много слоёв…




**Слева (кодировки текста) – память**
**Справа (изменения вектора ответа) – рекуррентная часть**

**Обращаемся в память и корректируем ответ…**

**Память выдаёт**

$$\sum_i p_i c_i$$

**веса ~ softmax релевантности**

# Memory Network «MemN2N» – использование памяти

**Ещё фишки…**

**1. Шум как регуляризация**

**2. Представление предложений**

**Проблема: предложение – вектор**

- **сумма кодировок слов**
- **position encoding (PE) – взвешенная сумма (для учёта порядка)**

**3. Аналогично учёт контекста событий… (что было ДО)**

**4. Темп обучения понижался вручную**
**(без момента и сокращения весов)**
**Обучено несколько сетей (разная инициализация).**
**Выбрана с наименьшей ошибкой…**

# Понимания языка (Language Understanding)

## Что такое «понимание языка»

## 1) умение автоматически генерировать «желаемый ответ»

**Когда ходят в школу?**

**Желаемые:**
- в детстве
- с сентября

**Не желаемые:**
- никогда
- вчера

**Что изображено на рисунке?**

**Желаемые:**
- бананы
- фрукты

**Не желаемые:**
- жёлтые объекты

# Моделирование языка (Language Modeling)

## учимся генерировать текст

**Насколько вероятно предложение**
**«кот поймал в мешок дровосека»**

**Unigram Modelling**

$$p(кот)\cdot p(поймал)\cdot p(в)\cdot p(мешок)\cdot p(дровосека)$$

**Bigram Modelling**

$$p(кот)\cdot p(поймал|кот)\cdot p(в|поймал)\cdot p(мешок|в)\cdot p(дровосека|мешок)$$

**Trigram Modelling**

$$p(кот)\cdot p(поймал|кот)\cdot p(в|кот,поймал)\cdot p(мешок|поймал,в)...$$

# Проблема

**в корпусе может не быть некоторых сочетаний**

**Сглаживание**

$$p(x_t \mid x_{t-n},\ldots,x_{t-1}) = \frac{\#(x_{t-n},\ldots,x_{t-1},x_t)+\alpha}{\#(x_{t-n},\ldots,x_{t-1})+\alpha\,|V|}$$

**Backoff (примерно так…)**

**при** $\#(x_{t-n},\ldots,x_{t-1}) = 0$

$$p(x_t \mid x_{t-n},\ldots,x_{t-1}) = \alpha(x_{t-n},\ldots,x_{t-1})\frac{\#(x_{t-n+1},\ldots,x_{t-1},x_t)}{\#(x_{t-n+1},\ldots,x_{t-1})}$$

**умножаем на некоторый «понижающий множитель»**

# Проблема

**Маленькое обобщение (Lack of Generalization)**

**(идти, в, сад), (идти, в, огород)**
**p(идти, в, парк) =?**

**Выход: моделирование языка с помощью НС**

# Параметрическое оценивание

$$p(x_t \mid x_{t-n},\ldots,x_{t-1}) = p_{x_t}(x_{t-n},\ldots,x_{t-1})$$

**пусть зависимость от n предыдущих**

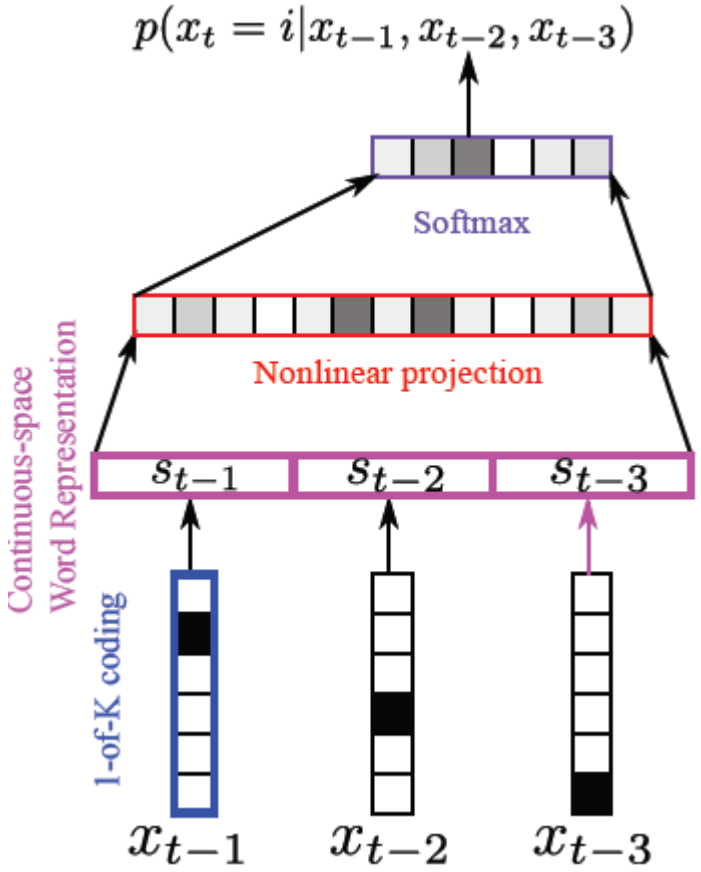
**ONE для слов**

$$s_j = W_{d\times|V|}x_j$$

$$h = \tanh(U_{d'\times nd}[s_{t-1},\ldots,s_{t-n}]+b)$$

$$y = V_{|V|\times d'}h + c$$

$$p(x_t = i \mid x_{t-n},\ldots,x_{t-1}) = \text{softmax}(y)$$

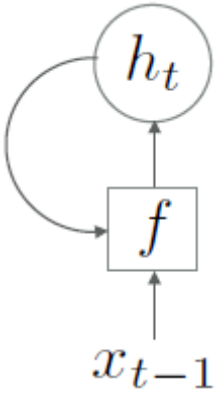


# Немарковские модели

$$p(x_t,\ldots,x_T)=\prod_{t=1}^{T} p(x_t \mid x_1,\ldots,x_{t-1})$$

**т.е. зависимость от всех слов предложения!**

**Как подавать на вход НС информацию разной длины?**




**Рекурсия**

$$h_0 = 0$$

$$h_t = f(x_{t-1}, h_{t-1})$$ **(внутренне состояние = память)**

$$p(x_t \mid x_1,\ldots,x_{t-1}) = g(h_t)$$

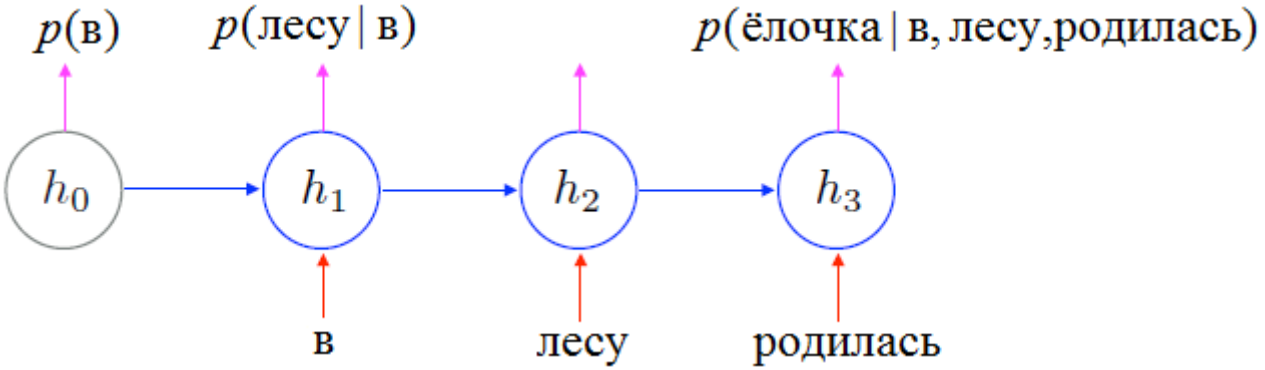
**f – transition function**

**g – output (readout) function**

# RNN-моделирование языка

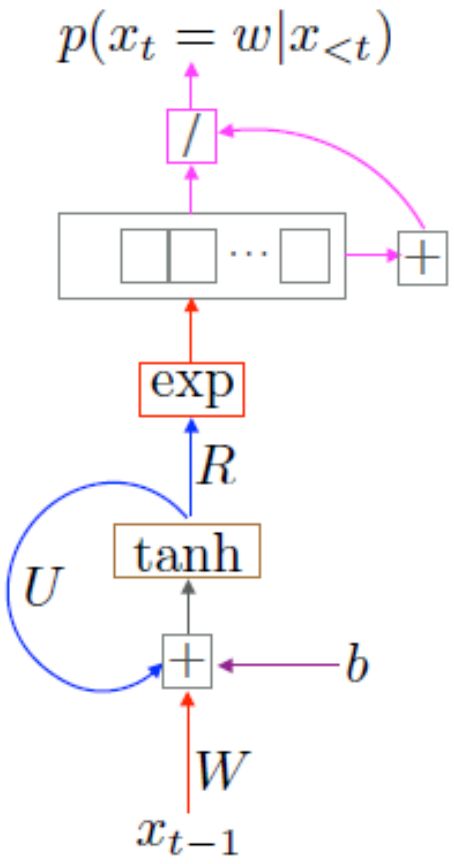
**p(в, лесу, родилась, ёлочка)**

$$h_0 = 0 \quad \Rightarrow \quad p(\text{в}) = g(h_0)$$
$$h_1 = f(h_0, \text{в}) \quad \Rightarrow \quad p(\text{лесу} \mid \text{в}) = g(h_1)$$
$$h_2 = f(h_1, \text{лесу}) \quad \Rightarrow \quad p(\text{родилась} \mid \text{в}, \text{лесу}) = g(h_2)$$
$$h_3 = f(h_2, \text{родилась}) \quad \Rightarrow \quad p(\text{ёлочка} \mid \text{в}, \text{лесу}, \text{родилась}) = g(h_3)$$
$$p(\text{в}, \text{лесу}, \text{родилась}, \text{ёлочка}) = g(h_0)g(h_1)g(h_2)g(h_3)$$




**рекуррентная сеть**

# RNN-моделирование языка




**Transition**

$$h_t = \tanh(W_{d\times|V|}x_{t-1} + U_{d\times d}h_{t-1} + b)$$

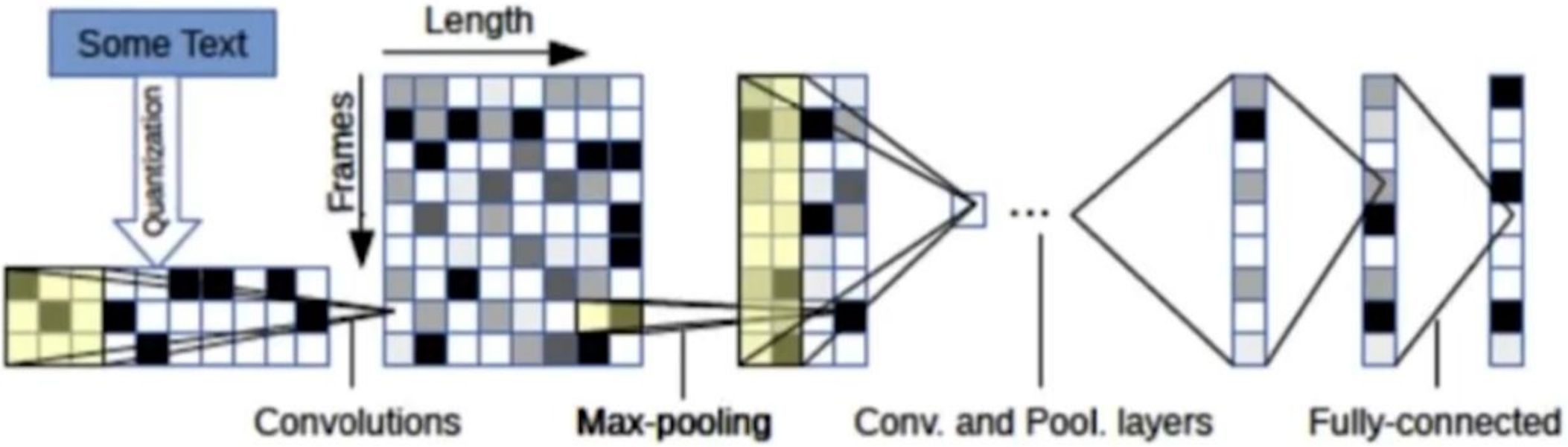
**Readout**

$$(p(x_t = w \mid x_{<t}))_{w=1}^{|V|} = g(h_t) = \text{softmax}(R_{|V|\times d}h_{t-1} + c)$$

**Обучение на выборке**

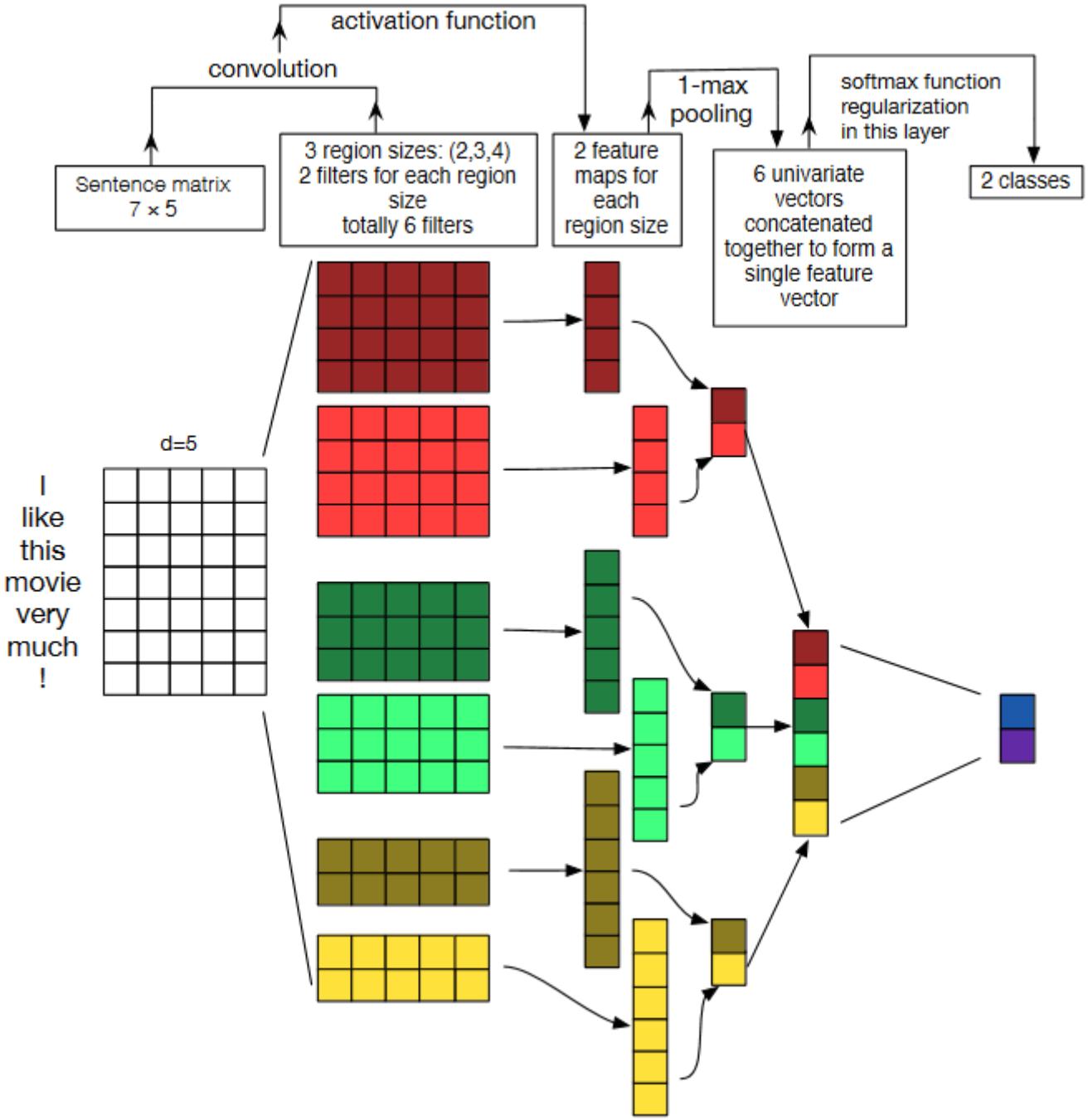
$$-\frac{1}{m}\sum_{i=1}^{m}\sum_{t=1}^{\text{len}(i)}\log p(x_t^{(i)} \mid x_1^{(i)},\ldots,x_{t-1}^{(i)}) \to \min$$

# Свёрточные модели для текста




## Дальше картинка из

Ye Zhang, Byron Wallace A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification https://arxiv.org/abs/1510.03820


activation function
convolution
1-max pooling
softmax function regularization in this layer
Sentence matrix 7 × 5
3 region sizes: (2,3,4) 2 filters for each region size totally 6 filters
2 feature maps for each region size
6 univariate vectors concatenated together to form a single feature vector
2 classes
d=5
I
like
this
movie
very
much
!

# Представление текстов

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

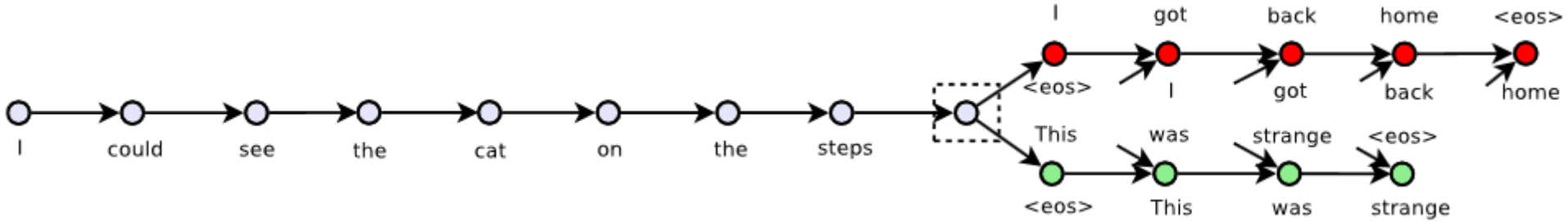
**текст ~ «среднее» векторов входящих слов**

## Distributed Memory Model of Paragraph Vectors (PV-DM)

**не будем подробно**

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

# Представление предложений: The skip-thoughts model




**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**

Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Antonio Torralba, Raquel Urtasun, Sanja Fidler Skip-Thought Vectors // https://arxiv.org/abs/1506.06726

# 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 |

# Представление слов/предложений/текстов: StarSpace

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

**Метод оперирует с объектами, которые описываются наборами признаков из фиксированного множества**

**Пример: предложение = набор слов**
**(или = набор n-грамм)**

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

**~ генерация позитивных и негативных пар**

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

# 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**

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

**сумма по слоям**

**можно затачивать представление под конкретную задачу**

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

# Представление предложений: Deep Averaging Network (DAN)

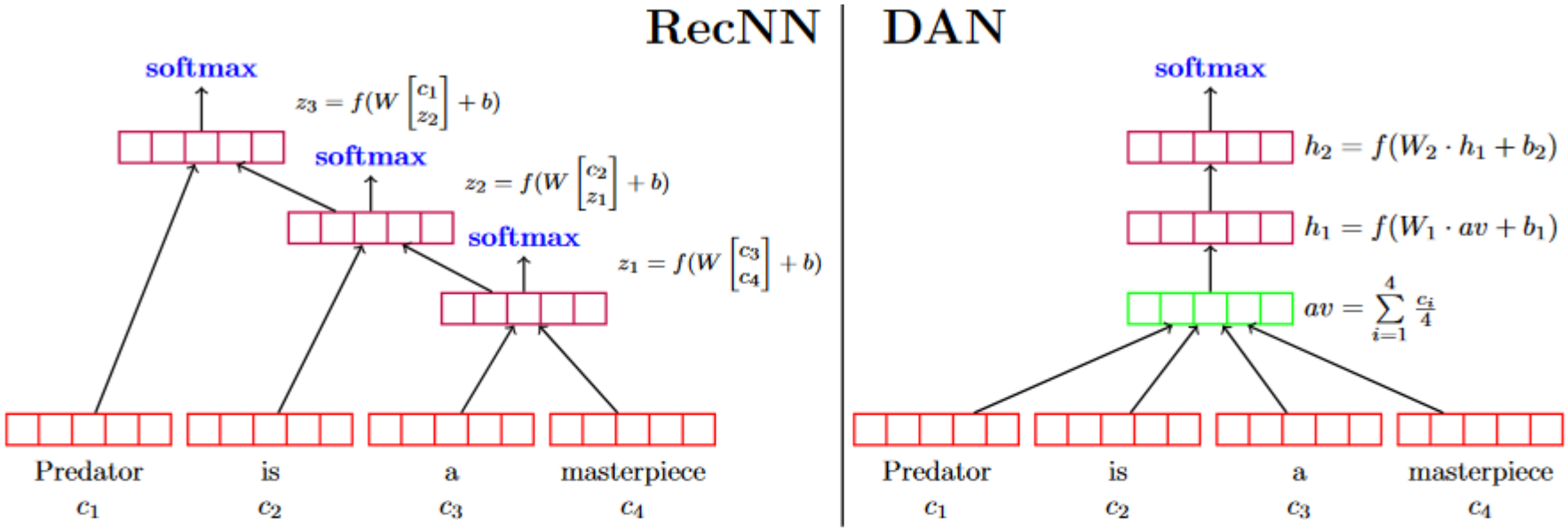



Figure 1: On the left, a **RecNN** is given an input sentence for sentiment classification. Softmax layers are placed above every internal node to avoid vanishing gradient issues. On the right is a two-layer **DAN** taking the same input. While the **RecNN** has to compute a nonlinear representation (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…**

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

| 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

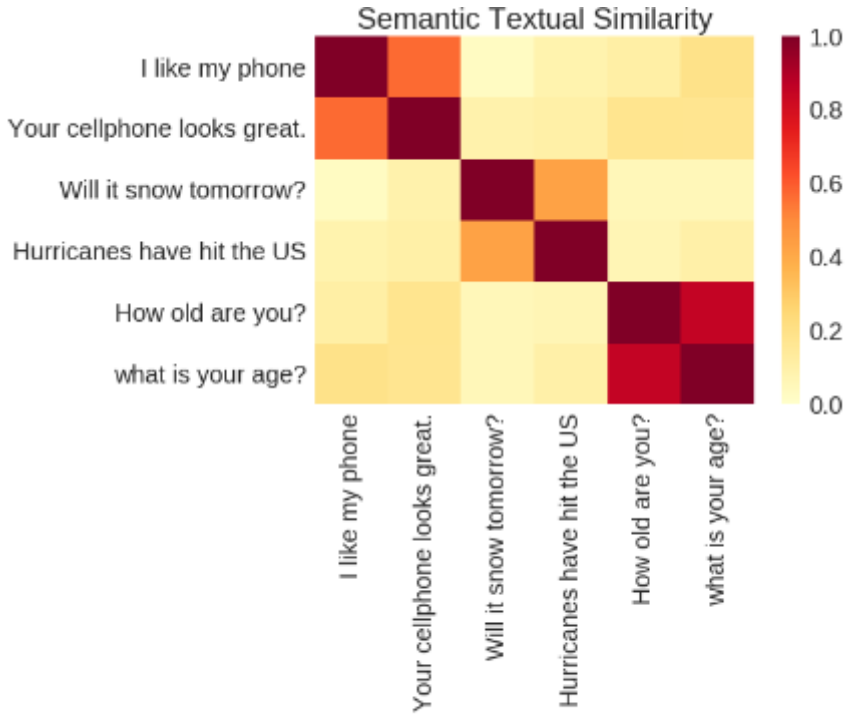



Figure 1: Sentence similarity scores using embeddings from the universal sentence encoder.

**использовали 1) Transformer 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

# Ещё подходы

## Чем проще агрегация кодировок слов, тем нехуже

Dinghan Shen, Guoyin Wang, Wenlin Wang, Martin Renqiang Min, Qinliang Su, Yizhe Zhang, Chunyuan Li, Ricardo Henao, Lawrence Carin Baseline Needs More Love: On Simple Word-Embedding-Based Models and Associated Pooling Mechanisms // https://arxiv.org/abs/1805.09843

## Обзор (полный, хороший)

Christian S. Perone, Roberto Silveira, Thomas S. Paula Evaluation of sentence embeddings in downstream and linguistic probing tasks // https://arxiv.org/abs/1806.06259

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 |

# Как вкладываются предложения: общий подход

**Вложение предложения ищется в виде**

$$h = f_\theta(e_1,\ldots,e_n)$$

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

| | |
|---|---|
| **IferSent** | $\max(\text{MiLSTM}(e_1,\ldots,e_n))$ **Обучаем предсказывая entailment, neutral or contradictive. cross-entropy** |
| **SkipThought** | $\text{GRU}_n(e_1,\ldots,e_n)$ **Декодируем следующее и предыдущее negative log-likelihood** |
| **Случайный кодировщик – не сильно хуже!** | **John Wieting, Douwe Kiela No Training Required: Exploring Random Encoders for Sentence Classification https://arxiv.org/abs/1901.10444** |

# Transformer

https://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova Bert: Pre-training of deep bidirectional transformers for language understanding. 2018.

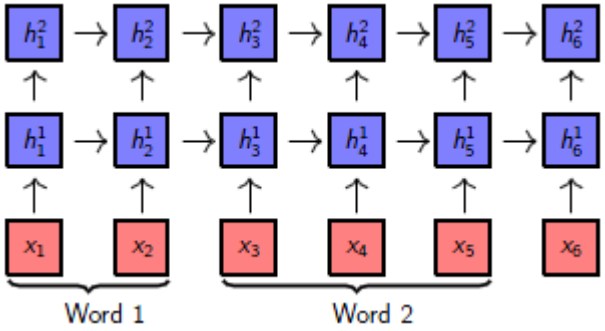
Emma Strubell, Patrick Verga, Daniel Andor, David Weiss, Andrew McCallum Linguistically-Informed Self-Attention for Semantic Role Labeling // https://arxiv.org/abs/1804.08199

# Hierarchical Multiscale Recurrent Neural Networks

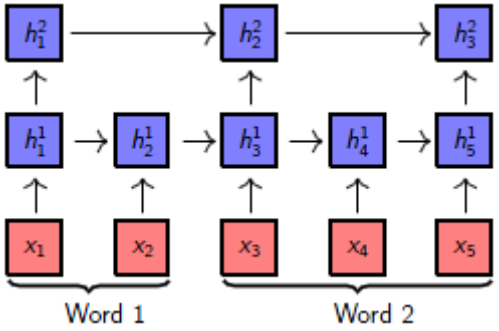
**У текстов структура на разных масштабах:**

**буквы → слова → фразы → предложения → абзацы**

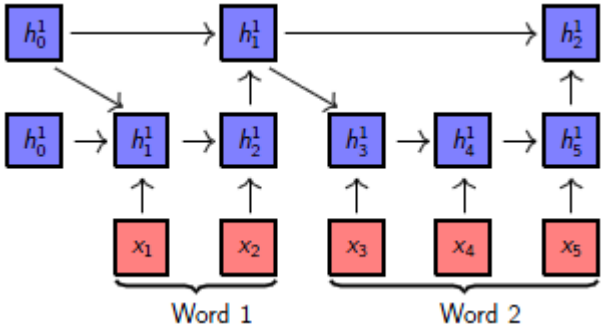
## Stacked RNN



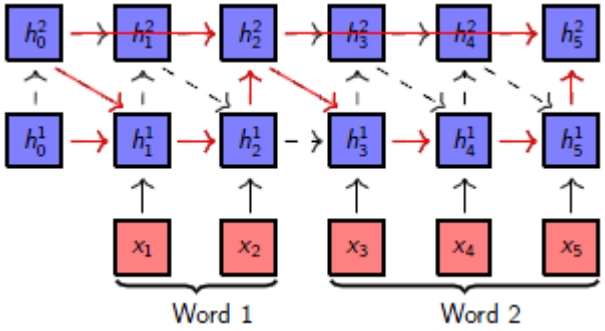

## Clockwork RNN




## Boundary-aware RNN




## Hierarchical Multiscale RNN




Junyoung Chung, Sungjin Ahn, Yoshua Bengio «Hierarchical Multiscale Recurrent Neural Networks», 2017 // https://arxiv.org/abs/1609.01704

# Hierarchical Multiscale Recurrent Neural Networks

**+ вычислительная эффективность (верхние слои проще)**

**+ меньше изменений $\Rightarrow$ лучше распространение информации**

**– сеть теперь не дифференцируема**

- **можно использовать Хэвисайда во время прямого распространения и игнорировать порог во время обратного**
- **можно склон делать всё более крутым**

# Результат 2019 – GPT2

https://blog.openai.com/better-language-models/

https://d4mucfpksywv.cloudfront.net/better-language-models/language_models_are_unsupervised_multitask_learners.pdf

**1.5 млрд параметров**

**Transformer**

**SOTA 7 из 8 задач (zero-shot setting)**

**обучение – новый датасет «WebText»**

**~ 1 млн web-страниц / 45 млн ссылок / 8 млн. документов 40Гб ???**

**ссылки с Reddit ≥3 кармы (т.е. отбором человека)**

**удалили Wiki ! (чтобы тестировать на других датасетах)**

**экстракторы текстов:**

**Dragnet (Peters & Lecocq, 2013) and Newspaper (https://github.com/codelucas/newspaper)**

# Результат 2019 – GPT2

**Задачи**

- **question answering**
- **machine translation**
- **reading comprehension**
- **summarization**

**В основе – Language modeling**

$$p(x) = \prod_{i=1}^{n} p(s_n | s_1, ..., s_{n-1})$$

**Современная оценка таких вероятностей:**
**self-attention architectures ~ Transformer (Vaswani et al., 2017)**

**p(output |input, task)**

# Результат 2019 – GPT2

**«Task»**

**~ специальная архитектура (encoders/decoders Kaiser et al., 2017)**
**~ специальные алгоритмы (inner/outer loop optimization framework of MAML Finn et al., 2017)**
**~ с помощью языка MQAN – McCann et al. (2018):**

**«переведи …»**

**«ответь на вопрос …»**

**« TL;DR:»**

**без дообучения с учителем на специализированных данных!**
**zero-shot task transfer**

# Результат 2019 – GPT2

## Предобработка

lower-casing
tokenization
out-of-vocabulary tokens
Unicode → UTF-8

**тут использована:**
**Byte Pair Encoding (BPE) (Sennrich et al., 2015)**
**кодируем частые слова и буквы (из которых состоят редкие слова)**

Sennrich, R., Haddow, B., and Birch, A. Neural machine translation of rare words with subword units // arXiv:1508.07909, 2015

# Результат 2019 – GPT2

## продолжение OpenAI GPT model (Radford et al., 2018)

| Parameters | Layers | $d_{model}$ |
|---|---|---|
| 117M | 12 | 768 |
| 345M | 24 | 1024 |
| 762M | 36 | 1280 |
| 1542M | 48 | 1600 |

*Table 2.* Architecture hyperparameters for the 4 model sizes.

**что нового**

**Layer normalization → вход каждого под-блока**

**Layer normalization → после self-attention-блока**

**другая инициализация**

**vocabulary = 50,257**

**context size = 1024**

**batchsize = 512**

**Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser,Ł., and Polosukhin, I. Attention is all you need. In Advances in Neural Information Processing Systems, pp. 5998–6008, 2017.**

# Результат 2019 – GPT2

**Language Models are Unsupervised Multitask Learners**

| | LAMBADA (PPL) | LAMBADA (ACC) | CBT-CN (ACC) | CBT-NE (ACC) | WikiText2 (PPL) | PTB (PPL) | enwik8 (BPB) | text8 (BPC) | WikiText103 (PPL) | 1BW (PPL) |
|---|---|---|---|---|---|---|---|---|---|---|
| SOTA | 99.8 | 59.23 | 85.7 | 82.3 | 39.14 | 46.54 | 0.99 | 1.08 | 18.3 | **21.8** |
| 117M | **35.13** | 45.99 | **87.65** | **83.4** | **29.41** | 65.85 | 1.16 | 1.17 | 37.50 | 75.20 |
| 345M | **15.60** | 55.48 | **92.35** | **87.1** | **22.76** | 47.33 | 1.01 | **1.06** | 26.37 | 55.72 |
| 762M | **10.87** | **60.12** | **93.45** | **88.0** | **19.93** | **40.31** | **0.97** | **1.02** | 22.05 | 44.575 |
| 1542M | **8.63** | **63.24** | **93.30** | **89.05** | **18.34** | **35.76** | **0.93** | **0.98** | **17.48** | 42.16 |

*Table 3.* Zero-shot results on many datasets. No training or fine-tuning was performed for any of these results. PTB and WikiText-2 results are from (Gong et al., 2018). CBT results are from (Bajgar et al., 2016). LAMBADA accuracy result is from (Hoang et al., 2018) and LAMBADA perplexity result is from (Grave et al., 2016). Other results are from (Dai et al., 2019).