

• Вероятностные языковые модели •
Лекция 3.
Модели локального контекста

Константин Вячеславович Воронцов
k.vorontsov@iaai.msu.ru

Этот курс доступен на странице вики-ресурса
<http://www.MachineLearning.ru/wiki>
«Вероятностные языковые модели (курс лекций, К.В.Воронцов)»

ВМК МГУ • 16 марта 2026

- 1 Нейросетевые языковые модели**
 - Введение в искусственные нейронные сети
 - Модель машинного перевода
 - Модель кодировщик BERT
- 2 Тематическая модель локального контекста**
 - Модель Attentive ARTM
 - Быстрое вычисление векторов контекста
 - EM-алгоритм для Attentive ARTM
- 3 Сравнение тематических моделей с нейросетевыми**
 - Сравнение с моделью внимания и трансформером
 - Свёрточная сеть GCNN
 - Нейросетевая тематическая модель Contextual-Top2Vec

Общая постановка большинства задач машинного обучения

Дано: X — пространство объектов

$X^\ell = \{x_1, \dots, x_\ell\} \subset X$ — обучающая выборка (training sample)

$a(x, w)$, $a: X \times W \rightarrow Y$ — параметрическая модель, гипотеза

Найти $w \in W \subseteq \mathbb{R}^N$ — вектор параметров модели $a(x, w)$

Критерий \min регуляризованного эмпирического риска
(Regularized ERM — Empirical Risk Minimization):

$$Q(w, X^\ell) = \frac{1}{\ell} \sum_{i=1}^{\ell} \mathcal{L}(w, x_i) + \tau \mathcal{R}(w) \rightarrow \min_w$$

$\mathcal{L}(w, x)$ — функция потерь (loss function),

тем больше, чем хуже ответ модели $a(x, w)$ на объекте x

- $\mathcal{L}(a, y) = (a - y)^2$ в задачах регрессии, $Y = \mathbb{R}$
- $\mathcal{L}(a, y) = [a \neq y]$ в задачах классификации, $|Y| < \infty$

$\mathcal{R}(w)$ — регуляризатор, не прецедентные требования к модели

Градиентный метод минимизации эмпирического риска

$$Q(w) = \frac{1}{\ell} \sum_{i=1}^{\ell} \mathcal{L}(w, x_i) + \tau \mathcal{R}(w) \rightarrow \min_w$$

Метод *градиентного спуска*:

$w^{(0)}$:= начальное приближение;

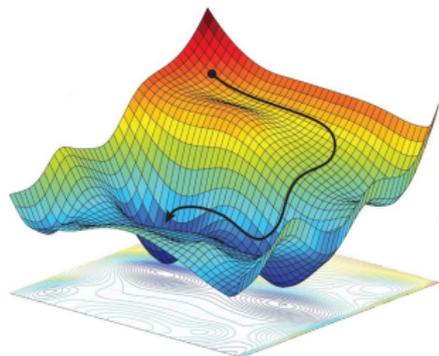
$w^{(t+1)} := w^{(t)} - h \nabla Q(w^{(t)})$

где $\nabla Q(w) = \left(\frac{\partial Q(w)}{\partial w_j} \right)_{j=1}^N$ — *вектор градиента*,

h — *градиентный шаг*, называемый также *темпом обучения*

$w^{(t+1)} := w^{(t)} - h \left(\frac{1}{\ell} \sum_{i=1}^{\ell} \nabla \mathcal{L}(w^{(t)}, x_i) + \tau \nabla \mathcal{R}(w^{(t)}) \right)$

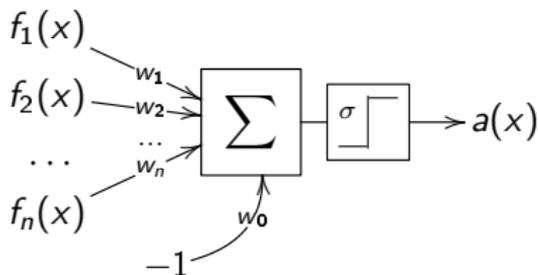
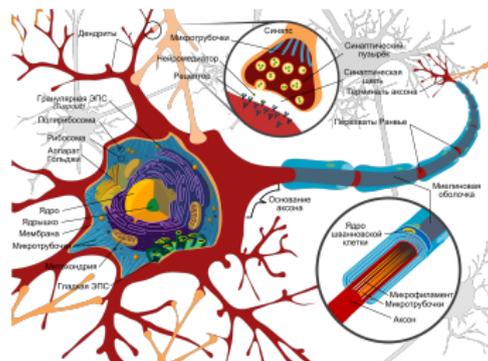
Идея ускорения сходимости: брать случайное подмножество слагаемых, чаще обновляя вектор w , или вообще один объект — *метод стохастического градиента* (Stochastic Gradient, SG)



Искусственный нейрон — линейная модель классификации

Линейная модель нейрона (1943):

$$a(x, w) = \sigma \left(\sum_{j=1}^n w_j f_j(x) - w_0 \right)$$

 $f_j(x)$ — признаки объекта x w_j — веса признаков w_0 — порог активации $\sigma(z)$ — функция активацииУоррен
МакКаллок
(1898–1969)Вальтер
Питтс
(1923–1969)

Многослойный перцептрон (MultiLayer Perceptron, MLP)

Архитектура сети: H_l — число нейронов в l -м слое, $l = 1, \dots, L$

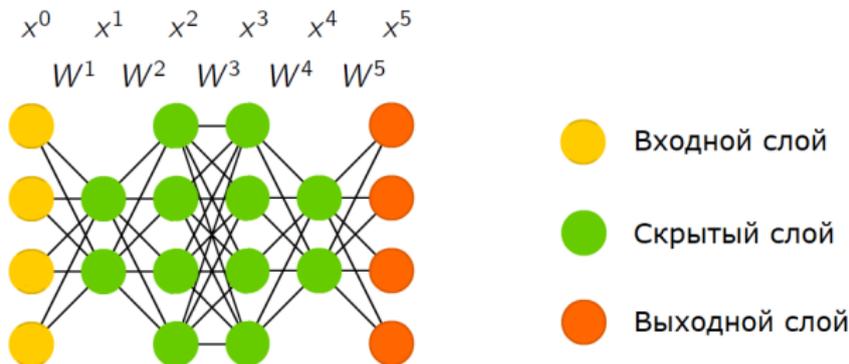
$x^0 \in \mathbb{R}^{n+1}$ — вектор признаков на входе сети, $x_0^0 = -1$

$x^l \in \mathbb{R}^{H_l}$ — вектор «признаков» на выходе l -го слоя, $x_0^l = -1$

$x^L \in \mathbb{R}^{H_L}$ — вектор на выходе сети

W^l — матрица весов l -го слоя, размера $(H_{l-1}+1) \times H_l$

$x^l = \sigma^l(W^l x^{l-1})$ — вычисление сети по слоям $l = 1, \dots, L$

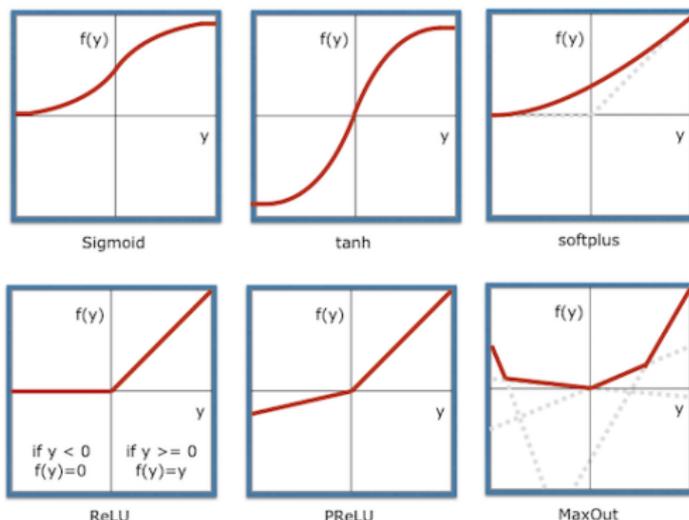


Функции активации

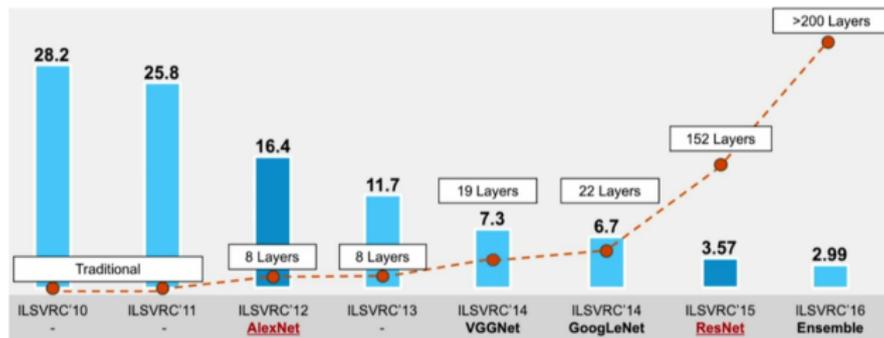
Функции $\sigma(y) = \frac{1}{1+e^{-y}}$ и $\text{th}(y) = \frac{e^y - e^{-y}}{e^y + e^{-y}}$ могут приводить к затуханию градиентов или «параличу сети»

Функция положительной срезки (Rectified Linear Unit, ReLU)

$$\text{ReLU}(y) = \max\{0, y\}; \quad \text{PReLU}(y) = \max\{0, y\} + \alpha \min\{0, y\}$$



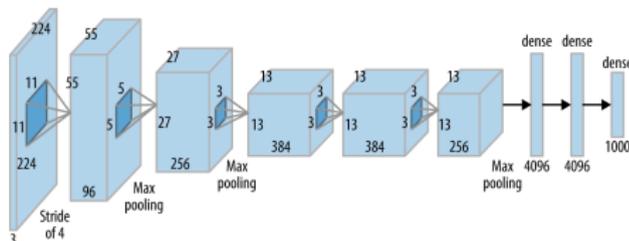
Глубокие свёрточные сети для классификации изображений



Старт в 2009. Человеческий уровень ошибок 5% пройден в 2015

Свёрточная сеть **AlexNet**:

- + ReLU + Dropout
- + 60M параметров
- + пополнение выборки
- + подбор размеров слоёв
- + GPU



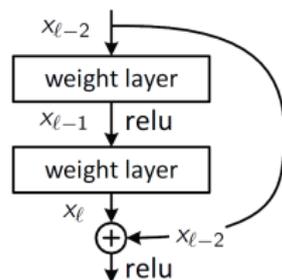
Krizhevsky A. et al. ImageNet classification with deep convolutional neural networks. 2012.

ResNet: остаточная нейронная сеть (Residual NN)

Сквозная связь (skip connection) слоя l
с предшествующим слоем $l - d$:

$$x_l = \sigma(Wx_{l-1}) + x_{l-d}$$

Слой l выучивает не новое векторное
представление x_l , а его приращение $x_l - x_{l-d}$



- Приращения более устойчивы \Rightarrow улучшается сходимость
- Появляется возможность увеличивать число слоёв
- Обобщение — Highway Networks:

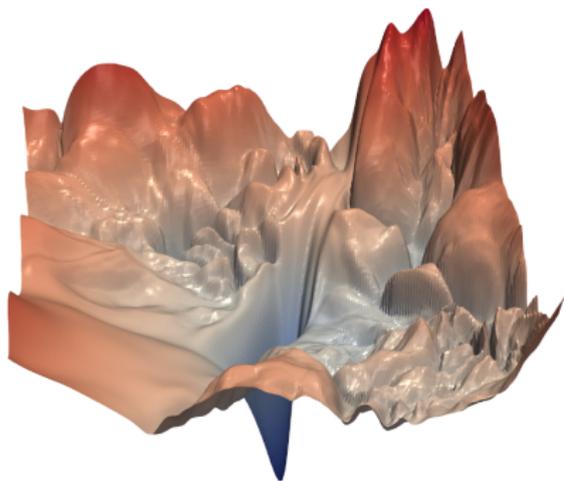
$$x_l = \sigma(Wx_{l-1}) \underbrace{\tau(W'x_{l-1})}_{\text{transform gate}} + x_{l-d} \underbrace{(1 - \tau(W'x_{l-1}))}_{\text{carry gate}}$$

Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. Deep Residual Learning for Image Recognition. 2015

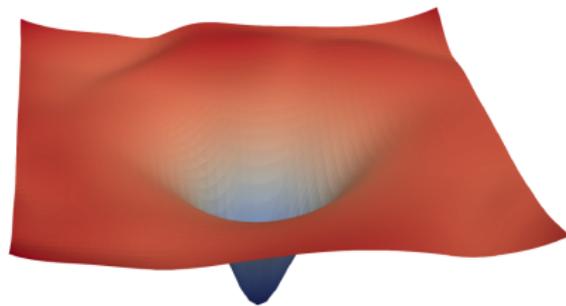
R.K.Srivastava, K.Greff, J.Schmidhuber. Highway Networks. 2015

ResNet: визуализация оптимизационного критерия

Сквозные связи (skip connection) упрощают оптимизируемый критерий, устраняя локальные экстремумы и седловые точки:



without skip connections



with skip connections

Hao Li et al. Visualizing the Loss Landscape of Neural Nets. 2018

Резюме по краткому введению в глубокое обучение

- Глубокая нейросеть — это обучаемая векторизация объектов

$$\frac{1}{\ell} \sum_{i=1}^{\ell} \mathcal{L}(w, f(x_i, w')) + \tau \mathcal{R}(w) \rightarrow \min_{w, w'}$$

где $f(x, w')$, $f: X \times W' \rightarrow \mathbb{R}^d$ — модель векторизации

- Обучение методом обратного распространения BackProp — это SG с быстрым вычислением градиента суперпозиции
- Возможности сети определяются её *архитектурой* — сколько в ней слоёв и как в них связаны нейроны
- Особый тип векторизации — автокодировщики:

$$\sum_{i=1}^{\ell} \mathcal{L}(g(f(x_i, \alpha), \beta), x_i) \rightarrow \min_{\alpha, \beta}$$

$f: X \rightarrow Z$ кодирует x в кодовый вектор $z = f(x, \alpha)$

$g: Z \rightarrow X$ декодирует, реконструирует $\hat{x} = g(z, \beta)$

Трасформер для машинного перевода

Трасформер (transformer) — это нейросетевая архитектура для трансформации векторов слов с учётом их контекста

Схема преобразований данных в машинном переводе:

- $S = (w_1, \dots, w_n)$ — слова предложения на входном языке
↓ обучаемая или пред-обученная векторизация слов
- $X = (x_1, \dots, x_n)$ — векторы слов входного предложения
↓ трансформер-кодировщик
- $Z = (z_1, \dots, z_n)$ — контекстно-зависимые векторы слов
↓ трансформер-декодировщик, похож на кодировщика
- $Y = (y_1, \dots, y_m)$ — векторы слов выходного предложения
↓ генерация слов из построенной языковой модели
- $\tilde{S} = (\tilde{w}_1, \dots, \tilde{w}_m)$ — слова предложения на выходном языке

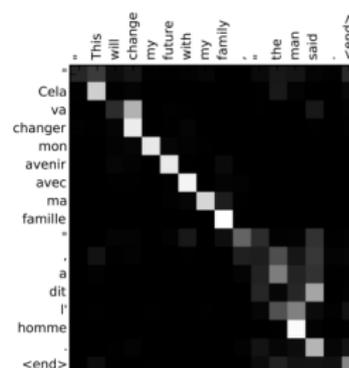
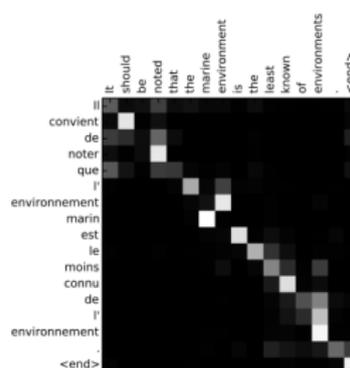
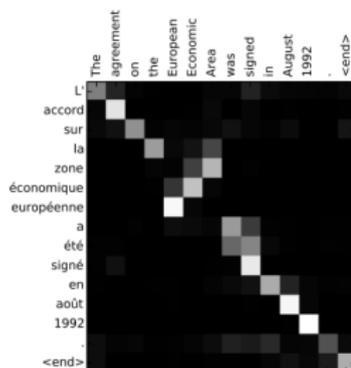
Vaswani et al. (Google) Attention is all you need. 2017.

Модели внимания для машинного перевода

$X = (x_1, \dots, x_n)$ — векторы слов входного предложения

$Y = (y_1, \dots, y_m)$ — векторы слов выходного предложения

Модель внимания оценивает матрицу семантического сходства $A_{ti} = a(x_i, y_t)$ — насколько входное слово x_i важно (требуется внимания) для обработки выходного слова y_t



Модель внимания Query–Key–Value

q — вектор-запрос для трансформации в вектор-контекст z
 $K = (k_1, \dots, k_n)$ — векторы-ключи, сравниваемые с запросом
 $X = (x_1, \dots, x_n)$ — векторы-значения, образующие контекст
 Модель внимания — трёхслойная сеть, вычисляющая z как выпуклую комбинацию векторов x_i , релевантных запросу q :

$$z = \text{Attn}(q, K, X) = \sum_i x_i \text{SoftMax}_i a(k_i, q),$$

где $a(k, q)$ — оценка релевантности ключа k запросу q ,
 например $a(k, q) = k^T q$ или $k^T W q$ с матрицей параметров W

Модель внутреннего внимания (самовнимания, self-attention):

$$z_i = \text{Attn}(W_q x_i, W_k X, W_v X)$$

трансформирует входную последовательность $X = (x_1, \dots, x_n)$
 в выходную последовательность векторов контекста (z_1, \dots, z_n)

Архитектура трансформера-кодировщика

1. Добавляются позиционные векторы p_i :

$$h_i = x_i + p_i, \quad H = (h_1, \dots, h_n) \quad \begin{array}{l} d = \dim x_i, p_i, h_i = 512 \\ \dim H = 512 \times n \end{array}$$

2. J голов самовнимания:

$$h_i^j = \text{Attn}(W_q^j h_i, W_k^j H, W_v^j H) \quad \begin{array}{l} j = 1, \dots, J = 8 \\ \dim h_i^j = 64 \\ \dim W_q^j, W_k^j, W_v^j = 64 \times 512 \end{array}$$

3. Конкатенация (multi-head attention):

$$h_i' = \text{MH}_j(h_i^j) \equiv [h_i^{j_1} \dots h_i^{j_J}] \quad \dim h_i' = 512$$

4. Сквозная связь + нормировка уровня:

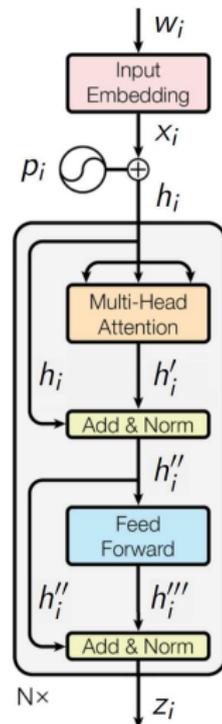
$$h_i'' = \text{LN}(h_i' + h_i; \mu_1, \sigma_1) \quad \dim h_i'', \mu_1, \sigma_1 = 512$$

5. Полносвязная 2х-слойная сеть FFN:

$$h_i''' = W_2 \text{ReLU}(W_1 h_i'' + b_1) + b_2 \quad \begin{array}{l} \dim W_1 = 2048 \times 512 \\ \dim W_2 = 512 \times 2048 \end{array}$$

6. Сквозная связь + нормировка уровня:

$$z_i = \text{LN}(h_i''' + h_i''; \mu_2, \sigma_2) \quad \dim z_i, \mu_2, \sigma_2 = 512$$



Несколько дополнений и замечаний

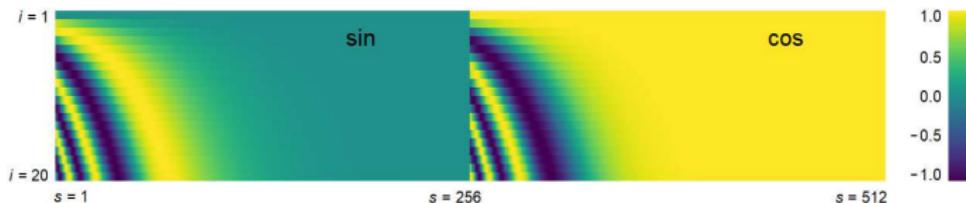
- $N = 6$ блоков $h_i \rightarrow \square \rightarrow z_i$ соединяются последовательно
- эмбединги слов $x_i \in \mathbb{R}^d$ — обучаемые или пред-обученные
- нормировка уровня (Layer Normalization), $x, \mu, \sigma \in \mathbb{R}^d$:

$$\text{LN}_s(x; \mu, \sigma) = \sigma_s \frac{x_s - \bar{x}}{\sigma_x} + \mu_s, \quad s = 1, \dots, d,$$

$\bar{x} = \frac{1}{d} \sum_s x_s$ и $\sigma_x^2 = \frac{1}{d} \sum_s (x_s - \bar{x})^2$ — среднее и дисперсия x

- Позиции слов i кодируются векторами $p_i, i = 1, \dots, n$; чем больше $|i - j|$, тем больше $\|p_i - p_j\|$, n не ограничено:

$$p_{is} = \sin(i 10^{-8} \frac{s}{d}), \quad p_{i, s + \frac{d}{2}} = \cos(i 10^{-8} \frac{s}{d}), \quad s = 1, \dots, \frac{d}{2}$$



Архитектура трансформера декодировщика

Авторегрессионный синтез последовательности:

$y_0 = \langle \text{BOS} \rangle$ — вектор символа начала;

для всех $t = 1, 2, \dots$:

1. Маскирование «данных из будущего»:

$$h_t = y_{t-1} + p_t; \quad H_t = (h_1, \dots, h_t)$$

2. Многомерное самовнимание:

$$h'_t = \text{LN} \circ \text{MH}_j \circ \text{Attn}(W_q^j h_t, W_k^j H_t, W_v^j H_t)$$

3. Многомерное внимание на кодировку Z :

$$h''_t = \text{LN} \circ \text{MH}_j \circ \text{Attn}(\tilde{W}_q^j h'_t, \tilde{W}_k^j Z, \tilde{W}_v^j Z)$$

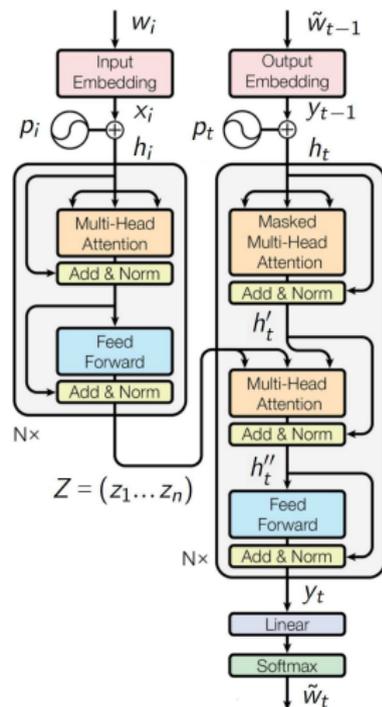
4. Двухслойная полносвязная сеть:

$$y_t = \text{LN} \circ \text{FFN}(h''_t)$$

5. Линейный предсказывающий слой:

$$p(\tilde{w}|t) = \text{SoftMax}_{\tilde{w}}(W_y y_t + b_y)$$

генерация $\tilde{w}_t = \arg \max_{\tilde{w}} p(\tilde{w}|t)$ **пока** $\tilde{w}_t \neq \langle \text{EOS} \rangle$



Vaswani et al. (Google) Attention is all you need. 2017.

Критерии обучения и валидации для машинного перевода

Критерий для обучения параметров нейронной сети W по обучающей выборке предложений S с переводом \tilde{S} :

$$\sum_{(S, \tilde{S})} \sum_{\tilde{w}_t \in \tilde{S}} \ln p(\tilde{w}_t | t, S, W) \rightarrow \max_W$$

Критерий оценивания моделей (недифференцируемые) по выборке пар предложений «перевод S , эталон S_0 »:

BiLingual Evaluation Understudy:

$$\text{BLEU} = \min\left(1, \frac{\sum \text{len}(S)}{\sum \text{len}(S_0)}\right) \text{mean}_{(S_0, S)} \left(\prod_{n=1}^4 \frac{\#n\text{-грамм из } S, \text{ входящих в } S_0}{\#n\text{-грамм в } S} \right)^{\frac{1}{4}}$$

Word Error Rate:

$$\text{WER} = \text{mean}_{(S_0, S)} \left(\frac{\#вставок + \#удалений + \#замен}{\text{len}(S)} \right)$$

Vaswani et al. (Google) Attention is all you need. 2017.

BERT (Bidirectional Encoder Representations from Transformers)

Трансформер BERT — это кодировщик без декодировщика, предобучаемый на большой текстовой коллекции для решения широкого класса задач автоматической обработки текста

Схема преобразования данных в задачах NLP:

- $S = (w_1, \dots, w_n)$ — токены предложения входного текста
↓ обучение эмбедингов вместе с трансформером
- $X = (x_1, \dots, x_n)$ — эмбединги токенов входного предложения
↓ трансформер кодировщика
- $Z = (z_1, \dots, z_n)$ — трансформированные эмбединги
↓ дообучение на конкретную задачу
- Y — выходной текст / разметка / классификация и т.п.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova (Google AI Language)
BERT: pre-training of deep bidirectional transformers for language understanding. 2019.

Критерий MLM (masked language modeling) для обучения BERT

Критерий маскированного языкового моделирования MLM, строится автоматически по текстам (self-supervised learning):

$$\sum_S \sum_{i \in M(S)} \ln p(w_i | i, S, W) \rightarrow \max_W,$$

где $M(S)$ — подмножество маскированных токенов из S ,

$$p(w | i, S, W) = \text{SoftMax}_{w \in V}(W_z z_i(S, W_T) + b_z)$$

— языковая модель, предсказывающая i -й токен предложения S ;

$z_i(S, W_T)$ — контекстный эмбединг i -го токена предложения S на выходе трансформера-кодировщика с параметрами W_T ;

$W = (W_T, W_z, b_z)$ — все параметры языковой модели

Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova (Google AI Language)
BERT: pre-training of deep bidirectional transformers for language understanding. 2019.

Критерий NSP (next sentence prediction) для обучения BERT

Критерий предсказания связи между предложениями NSP, строится автоматически по текстам (self-supervised learning):

$$\sum_{(S, S')} \ln p(y_{SS'} | S, S', W) \rightarrow \max_W,$$

где $y_{SS'} = [за S следует S']$ — классификация пары предложений,

$$p(y|S, S', W) = \text{SoftMax}_{y \in \{0,1\}}(W_y \text{th}(W_s z_0(S, S', W_T) + b_s) + b_y)$$

— вероятностная модель бинарной классификации пар (S, S') ,
 $z_0(S, S', W_T)$ — контекстный эмбединг токена $\langle \text{CLS} \rangle$ для пары предложений, записанной в виде $\langle \text{CLS} \rangle S \langle \text{SEP} \rangle S' \langle \text{SEP} \rangle$

Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova (Google AI Language)
 BERT: pre-training of deep bidirectional transformers for language understanding. 2019.

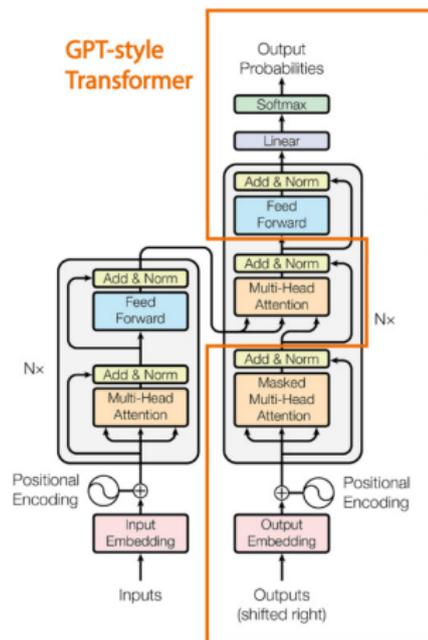
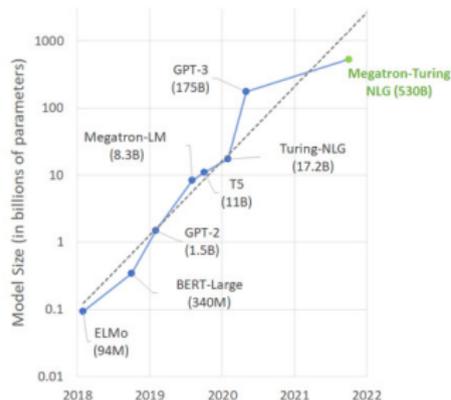
Ещё несколько замечаний про трансформеры

- **Fine-tuning:** для дообучения на задаче задаётся модель $f(Z(S, W_T), W_f)$, выборка $\{S\}$ и критерий $\mathcal{L}(S, f) \rightarrow \max$
- **Multi-task learning:** для дообучения на наборе задач $\{t\}$ задаются модели $f_t(Z(S, W_T), W_t)$, выборки $\{S\}_t$ и сумма критериев $\sum_t \lambda_t \sum_S \mathcal{L}_t(S, f_t) \rightarrow \max$
- *GLUE, SuperGLUE, Russian SuperGLUE, MERA, SLAVA* — наборы тестовых задач на понимание и генерацию языка
- Трансформеры обычно строятся не на словах, а на токенах, получаемых BPE (Byte-Pair Encoding) или WordPiece
- Первый трансформер: $N = 6$, $d = 512$, $J = 8$, весов 65M
- BERT_{BASE}, GPT1: $N = 12$, $d = 768$, $J = 12$, весов 110M
- BERT_{LARGE}: $N = 24$, $d = 1024$, $J = 16$, весов 340M

Генеративный преобученный трансформер (GPT, Open AI)

Generative Pre-trained Transformer:

- архитектура декодировщика остаётся (отличия не принципиальны)
- рост размера данных и моделей:



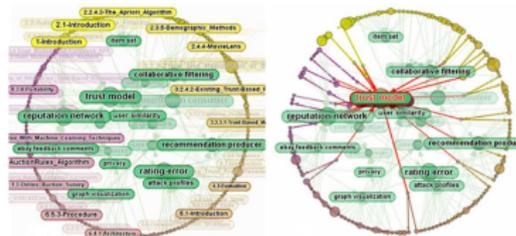
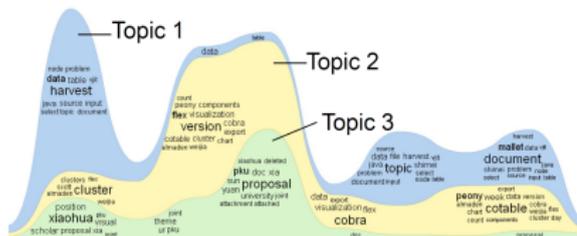
A.Radford et al. Improving language understanding by generative pre-training. 2018

A.Radford et al. Language models are unsupervised multitask learners. 2019 (GPT-2)

T.B.Brown et al. Language models are few-shot learners. 2020 (GPT-3)

Мотивации: что хотим от PTMs глядя на LLM (BERT, GPT)

- вместо «мешка слов» — последовательность w_1, \dots, w_n
- вместо документов — локальные контексты слов
- **ХОТИМ:** определять тематику любого фрагмента текста,
- быстро находить фрагменты, относящиеся к данной теме,
- в том числе фразы для суммаризации документа или темы,
- разделять документ на тематически однородные сегменты,
- визуализировать тематическую структуру документа.



Контекстная тематическая модель Attentive ARTM

Дано: коллекция текстовых документов, w_1, \dots, w_n
 $C_i \subset \{1, \dots, n\}$ — локальный контекст (окружение) термина w_i
 α_{ci} — коэффициент внимания, вес термина w_c в контексте C_i

Найти: $\phi_{tw} = p(t|w)$ — параметры тематической модели

$$p(w|C_i) = \sum_{t \in T} p(w|t)p(t|C_i) = \sum_{t \in T} p(t|w) \frac{p(w)}{p(t)} p(t|C_i)$$

$$p(t|C_i) \equiv \theta_{ti} = \sum_{c \in C_i} \alpha_{ci} p(t|w_c), \quad \sum_{c \in C_i} \alpha_{ci} = 1, \quad \alpha_{ci} \geq 0$$

Критерий: максимум \log правдоподобия с регуляризатором R :

$$\sum_{i=1}^n \ln \sum_{t \in T} \phi_{tw_i} \frac{p(w_i)}{p(t)} \sum_{c \in C_i} \alpha_{ci} \phi_{tw_c} + R(\Phi) \rightarrow \max_{\Phi}$$

EM-алгоритм для модели Attentive ARTM

EM-алгоритм: метод простой итерации для системы уравнений

$$p_{ti} \equiv p(t|C_i, w_i) = \operatorname{norm}_{t \in T} (\phi_{tw_i} \theta_{ti} / p(t)) \quad \theta_{ti} = \sum_{c \in C_i} \alpha_{ci} \phi_{tw_c}$$

$$N_{tw} = \sum_{i=1}^n q_{wi} \frac{p_{ti}}{\theta_{ti}} \quad q_{wi} = \sum_{c \in C_i} \alpha_{ci} [w_c = w]$$

$$\phi_{tw} = \operatorname{norm}_{t \in T} \left(n_{tw} + \phi_{tw} N_{tw} + \phi_{tw} \frac{\partial R}{\partial \phi_{tw}} \right) \quad n_{tw} = \sum_{i=1}^n p_{ti} [w_i = w]$$

Интерпретация вспомогательных переменных: n_{tw} — сколько раз терм w относится к теме t в коллекции q_{wi} — суммарный вес всех вхождений термина w в контекст C_i N_{tw} — суммарный вес термина w в контекстах темы t $p(t) = \frac{n_t}{n} = \frac{1}{n} \sum_{i=1}^n p_{ti}$ — «массовая» доля темы t в коллекции

Регуляризирующее воздействие локального контекста

Добавка N_{tw} похожа на регуляризатор R_0 т. ч. $\frac{\partial R_0}{\partial \phi_{tw}} = N_{tw}$,
можно домножить на τ , полагая по умолчанию $\tau = 0$

N_{tw} увеличивает $\phi_{tw} = p(t|w)$, если w часто встречается
в локальных контекстах C_i , где центральный терм w_i
относится к теме t с большой вероятностью:

$$N_{tw} = \sum_{i=1}^n q_{wi} \frac{p_{ti}}{\theta_{ti}} = \sum_{i=1}^n q_{wi} \frac{p(t|C_i, w_i)}{p(t|C_i)} = \sum_{i=1}^n q_{wi} \frac{p(w_i|t)}{p(w_i|C_i)}$$

N_{tw} похожа на дистрибутивные модели языка типа word2vec,
их тематические аналоги BitermTM, WordTM, WordNetworkTM,
регуляризаторы когерентности, сближающие ϕ_{tw} близких слов

N_{tw} повышает когерентность \Rightarrow интерпретируемость тем

N_{tw} вычисляется за $O(k^2|T|)$, где k — длина контекста,
при этом остальные операции с документом занимают $O(k|T|)$

Частный случай: контекст равен документу, «мешок слов»

$I_d = [i_d^{\text{beg}}, \dots, i_d^{\text{end}}]$ — термы документа d в сквозной нумерации

$C_i = I_d \Leftrightarrow i \in I_d$ — локальный контекст = весь документ

$\alpha_{ci} = \frac{1}{n_d} [c \in I_d]$ — внимание равномерно по документу

Тогда $\theta_{ti} = \theta_{td}$, $q_{wi} = \frac{n_{dw}}{n_d} = \hat{p}(w|d)$ — не зависят от i ,

$N_{tw} = n_w$ — не зависит от t , $\phi_{tw}^* N_{tw} = n_{tw}$

$$p_{tdw} = \operatorname{norm}_{t \in T} (\phi_{tw} \theta_{td} / p(t)); \quad \theta_{td} = \sum_{w \in d} \frac{n_{dw}}{n_d} \phi_{tw};$$

$$\phi_{tw} = \operatorname{norm}_{t \in T} \left(2n_{tw} + \phi_{tw} \frac{\partial R}{\partial \phi_{tw}} \right); \quad n_{tw} = \sum_{d \in D} n_{dw} p_{tdw};$$

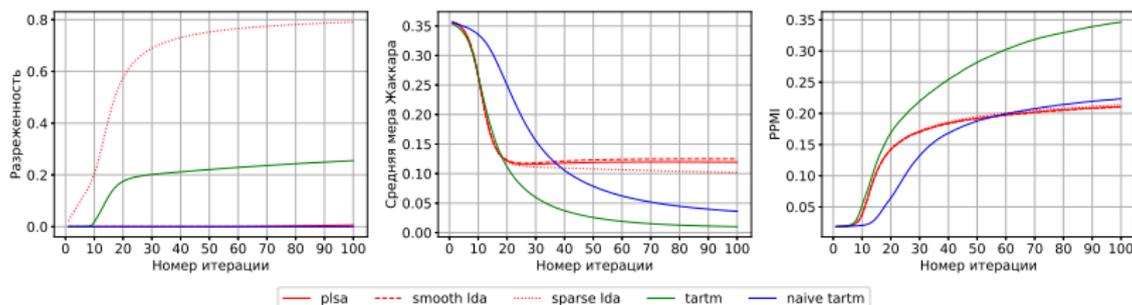
$$p(t) = \sum_{w \in W} \frac{n_{dw}}{n} p_{tdw}.$$

И.А.Ирхин, В.Г.Булатов, К.В.Воронцов. Аддитивная регуляризация тематических моделей с быстрой векторизацией текста, 2020.

Эксперимент. Проверка модифицированного EM-алгоритма

Коллекция NIPS, $|T| = 50$, модели:

- TARTM (Θ less ARTM) — модифицированный EM-алгоритм
- naive TARTM — одна итерация обычного EM-алгоритма



- TARTM очищает темы от общеупотребительных слов,
- улучшает разреженность, различность и когерентность тем

И.А.Ирхин, В.Г.Булатов, К.В.Воронцов. Аддитивная регуляризация тематических моделей с быстрой векторизацией текста, 2020.

https://github.com/ilirhin/python_artm

Как быстро вычислять взвешенные средние по контексту

Два прохода по тексту — «слева направо» и «справа налево» для вычисления *экспоненциальных скользящих средних* (ЭСС):

$$\vec{p}(t|i) = \vec{\gamma}_i p(t|w_i) + (1 - \vec{\gamma}_i) \vec{p}(t|i-1), \quad i = 1, \dots, n, \quad \vec{\gamma}_1 = 1$$

$$\vec{p}(t|i) = \vec{\gamma}_i p(t|w_i) + (1 - \vec{\gamma}_i) \vec{p}(t|i+1), \quad i = n, \dots, 1, \quad \vec{\gamma}_n = 1$$

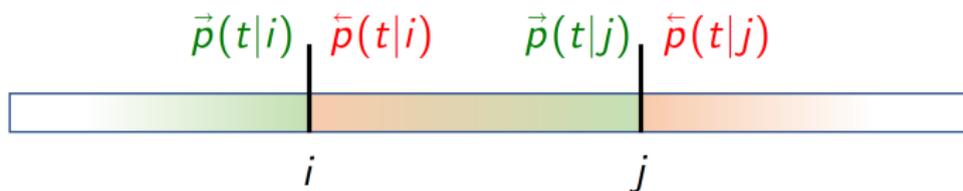
где $\vec{\gamma}_i, \vec{\gamma}_i$ — коэффициенты сглаживания в позиции i

Основное свойство: если $\gamma_i = \gamma$, то $\alpha_{ci} = \gamma(1 - \gamma)^{|i-c|}$

Несколько соображений, как распоряжаться выбором $\vec{\gamma}_i, \vec{\gamma}_i$:

- $\gamma_i \approx \frac{1}{h}$, где h — ширина окна, размер контекста
- $\gamma_i = 1$, если надо забыть контекст, сменить документ
- $\gamma_i = 0$, если надо проигнорировать терм
- γ_i можно умножать на оценку важности терма

Использование двунаправленных векторов контекста



Через двунаправленные тематические векторы определяется:

- $\vec{p}(t|i)$ — тематика левого контекста термина w_i
- $\vec{p}(t|i)$ — тематика правого контекста термина w_i
- $\frac{1}{2}(\vec{p}(t|i) + \vec{p}(t|i))$ — тематика двустороннего контекста w_i
- $p(t|i \dots j) = \frac{1}{2}(\vec{p}(t|i) + \vec{p}(t|j))$ — тематика сегмента $[i \dots j]$
- $\vec{p}(t|i) \approx \vec{p}(t|j)$ — однородность тематики сегмента $[i \dots j]$
- $\max_i \|\vec{p}(t|i) - \vec{p}(t|i)\|$ — граница i между сегментами
- при различных γ_i — короткие и длинные контексты

Аналогия с моделями языка GCNN, Attention, Transformer

EM-алгоритм для модели Attentive ARTM

Вход: текстовая коллекция, число тем $|T|$, параметры K, L ;

Выход: матрица Φ , векторы термов документов p_{ti} , $t \in T$, $i = 1, \dots, n$;

инициализация ϕ_{tw} ; $n_t := 1$ для всех $w \in W$, $t \in T$;

для всех итераций $k = 1..K$ (проходов по всей коллекции)

инициализация $(n_{tw}, N_{tw}, \tilde{n}_t) := 0$ для всех $w \in W$, $t \in T$;

для всех документов $d \in D$

$p_{ti} := \phi_{tw_i}$ для всех $t \in T$, $i \in I_d$;

для всех $l = 1..L$ (аналог L блоков внимания в трансформере)

$\theta_{ti} := \text{Attn}(p_{ti}: t \in T, i \in I_d)$;

$p_{ti} := \text{norm}_{t \in T}(p_{ti}\theta_{ti}/n_t)$ для всех $t \in T$, $i \in I_d$;

$q_{wi} := \text{Attn}([w_i = w]: w \in d, i \in I_d)$;

$N_{tw} := N_{tw} + q_{wi}p_{ti}/\theta_{ti}$ для всех $t \in T$, $w \in d$, $i \in I_d$;

$n_{tw_i} := n_{tw_i} + p_{ti}$; $\tilde{n}_t := \tilde{n}_t + p_{ti}$ для всех $t \in T$, $i \in I_d$;

$\phi_{tw} := \text{norm}_{t \in T}(n_{tw} + \frac{n_{tw}}{n_w} N_{tw} + \frac{n_{tw}}{n_w} \frac{\partial R}{\partial \phi_{tw}} \Big|_{\phi_{tw} = \frac{n_{tw}}{n_w}})$ для всех $t \in T$, $w \in W$;

$n_t := \tilde{n}_t$ для всех $t \in T$;

$y_{hi} := \text{Attn}(x_{hi}: h \in H, i \in I_d)$ означает $y_{hi} := \sum_{c \in C_i} \alpha_{ci} x_{hc}$

Модель внимания (self-attention) Query–Key–Value

Входные векторы слов (эмбединги)

$$X = (x_1, \dots, x_n) \in \mathbb{R}^T$$

трансформируются в векторы слов,
зависящие от контекстов C_i :

$$H = (h_1, \dots, h_n) \in \mathbb{R}^d$$

Модель внимания (self-attention):

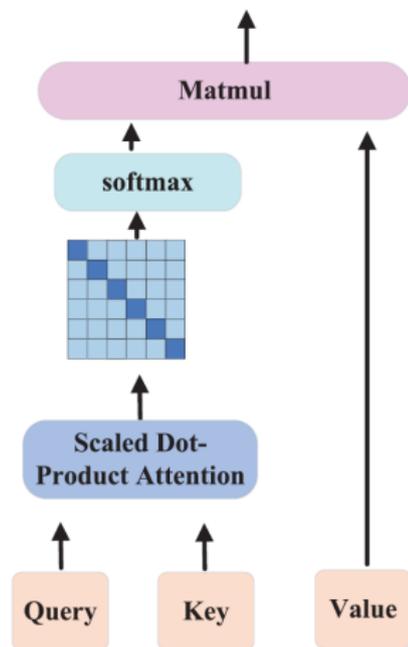
$$h_i = \sum_{c \in C_i} W_v x_c \text{SoftMax}_{c \in C_i} \langle W_k x_c, W_q x_i \rangle$$

$W_v x_c$ — вектор-значение (value)

$W_k x_c$ — вектор-ключ (key)

$W_q x_i$ — вектор-запрос (query)

W_q, W_k, W_v — обучаемые параметры



Аналогия Attentive ARTM с моделью само-внимания

Контекстный тематический вектор на выходе E-шага:

$$p(t|C_i, w_i) \equiv p_{ti} = \operatorname{norm}_{t \in T} \left(\sum_{c \in C_i} \phi_{tw_c} \alpha_{ci} \frac{1}{p(t)} \phi_{tw_i} \right)$$

Контекстный вектор на выходе модели само-внимания:

$$h_i = \sum_{c \in C_i} W_v x_c \alpha_{ci} = \sum_{c \in C_i} W_v x_c \operatorname{SoftMax}_{c \in C_i} \langle W_k x_c, W_q x_i \rangle$$

Сходство:

- вектор термина w_i трансформируется в контекстный вектор
- путём усреднения векторов ϕ_{w_c} из контекста термина w_i ,
- наиболее (семантически) схожих с вектором термина w_i

Отличия локализованного E-шага:

- адамарово умножение вектора ϕ_{w_c} на вектор-фильтр ϕ_{w_i}
- нет обучаемых матриц W_q, W_k, W_v как у модели внимания
- проецирование итогового вектора на единичный симплекс

Аналогия локализованного E-шага с моделью трансформера

Один проход документа аналогичен модели внимания:

— для каждого $d \in D$, для каждой позиции $i = 1, \dots, n_d$
вычисляются 5 тематических векторов, связанных с термом w_i :

$\phi_{tw_i} = p(t|w_i)$ — бесконтекстный вектор терма

$\vec{p}(t|i)$, $\bar{p}(t|i)$ — векторы левого и правого контекста

$\theta_{ti} = \beta \vec{p}(t|i) + (1 - \beta) \bar{p}(t|i)$ — век. двустороннего контекста

$p_{ti} = \text{norm}_t(\phi_{w_it} \theta_{ti})$ — контекстный вектор терма $p(t|C_i, w_i)$

Несколько таких проходов аналогичны трансформеру:

контекстный вектор терма $p_{ti} = p(t|C_i, w_i)$ на следующем проходе берётся вместо его бесконтекстного вектора $\phi_{tw_i} = p(t|w_i)$

L итераций аналогичны L необучаемым блокам внимания

Свёрточная нейросеть GCNN (Gated Convolutional Network)

Входные векторы слов (эмбединги)

$$X = (x_1, \dots, x_n) \in \mathbb{R}^T$$

трансформируются в векторы слов, зависящие от контекстов C_i :

$$H = (h_1, \dots, h_n) \in \mathbb{R}^d$$

через адамарово произведение:

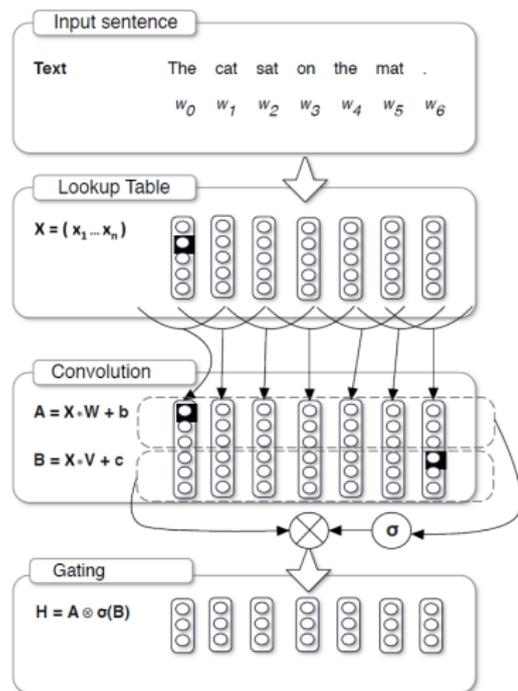
$$h_i = a_i \otimes \sigma(b_i), \text{ где}$$

$$a_i = \sum_{c \in C_i} W_c x_c \text{ — свёртка-контекст,}$$

$$b_i = \sum_{c \in C_i} V_c x_c \text{ — свёртка-фильтр,}$$

W_c, V_c — матрицы размера $d \times T$,
обучаемые параметры модели,

$$\sigma(x) = \frac{1}{1+e^{-x}} \text{ — функция сигмоида}$$



Yann N. Dauphin et al. Language modeling with gated convolutional networks, 2017.

Аналогия Attentive ARTM с моделью GCNN

Контекстный тематический вектор на выходе E-шага:

$$p(t|C_i, w_i) \equiv p_{ti} = \text{norm}_{t \in T} \left(\sum_{c \in C_i} \alpha_{ci} \phi_{tw_c} \frac{1}{p(t)} \phi_{tw_i} \right)$$

Контекстный вектор на выходе модели GCNN:

$$h_i = \left(\sum_{c \in C_i} W_c x_c \right) \otimes \sigma \left(\sum_{c \in C_i} V_c x_c \right)$$

Сходство:

- вектор термина w_i трансформируется в контекстный вектор
- путём усреднения векторов ϕ_{w_c} его контекста,
- семантически схожих с вектором термина w_i , фильтруемых адамаровым умножением на неотрицательный вектор

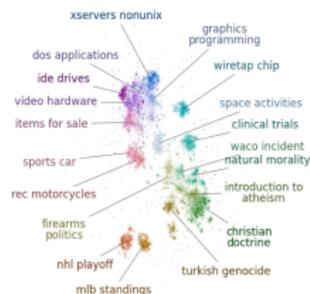
Отличия локализованного E-шага:

- нет обучаемых матриц W_c, V_c как у модели GCNN
- вектор-фильтр ϕ_{w_i} без усреднения по контексту C_i
- проецирование итогового вектора на единичный симплекс

Нейросетевая тематическая модель Contextual-Top2Vec

Вместо РТМ — сумма технологий:

- 1 векторизация токенов (Sentence-BERT)
- 2 векторизация предложений скользящим окном в 50 токенов (mean pooling)
- 3 понижение размерности векторов (UMAP)
- 4 иерархическая кластеризация (hDbSCAN) с автоматическим определением числа тем
- 5 иерархическое укрупнение тем слиянием мелких кластеров с ближайшими соседями (Top2Vec)
- 6 разбиение документа на монотематические сегменты
- 7 $p(t|d)$ = доля векторов данной темы в документе
- 8 именованые тем: поиск фраз, ближайших к центроиду темы



Dimo Angelov. Top2vec: Distributed representations of topics. 2020.

D. Angelov, D. Inkpen. Topic modeling: contextual token embeddings are all you need. 2024.

Нейросетевая тематическая модель Contextual-Top2Vec

Достоинства:

- модель BERT предобучена по большим внешним данным, поэтому качество тем не зависит от размера коллекции
- документ разбивается на монотематические сегменты
- автоматически определяется число тем (hDbSCAN, Top2Vec)
- тема описывается фразами, а не отдельными словами

Недостатки:

- это работает долго, особенно на больших коллекциях
- инкрементное добавление документов не предполагается

Сходство с Attentive ARTM:

- обработка локальных контекстов скользящим окном
- возможно разреживать $p(t|C_i)$ до монотематичности

Dimo Angelov. Top2vec: Distributed representations of topics. 2020.

D. Angelov, D. Inkpen. Topic modeling: contextual token embeddings are all you need. 2024.

Контекстная документная кластеризация (CDC)

n_{uw} — частота сочетания пары слов u, w в некотором окне

$p(u|w) = \frac{n_{uw}}{n_w}$ — контекст слова w

$H(w) = -\sum_u p(u|w) \log p(u|w)$ — энтропия контекста слова w

Узкий контекст — контекст с низкой энтропией, аналог темы, слова u , неслучайно часто встречающиеся рядом со словом w

Метод CDC — Contextual Document Clustering:

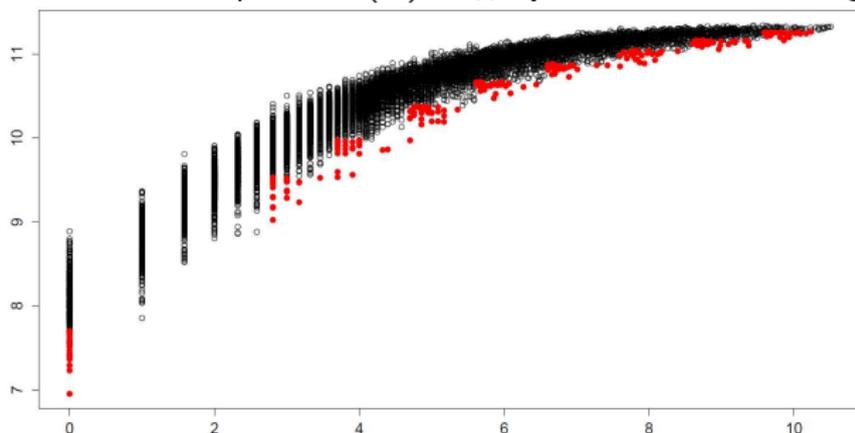
- 1 выделить «тематичные» слова с узкими контекстами
- 2 кластеризовать узкие контексты (найти кластеры-темы)
- 3 разбить документы на однородные сегменты (абзацы)
- 4 отнести каждый сегмент к ближайшей теме
- 5 $p(t|d) =$ доля сегментов темы t в документе

Vladimir Dobrynin, D.Patterson, N.Rooney. Contextual document clustering. 2004.
D.Patterson, N.Rooney, V.Dobrynin, M.Galushka. SOPHIA: A novel approach for textual case-based reasoning. 2005.

Выделение слов, имеющих узкие контексты

Оригинальный CDC: диапазон $\log_2 N_w$ разбивается на интервалы, в каждом интервале отбираются слова с наименьшими $H(w)$:

Зависимость энтропии $H(w)$ от документной частоты $\log_2 N_w$



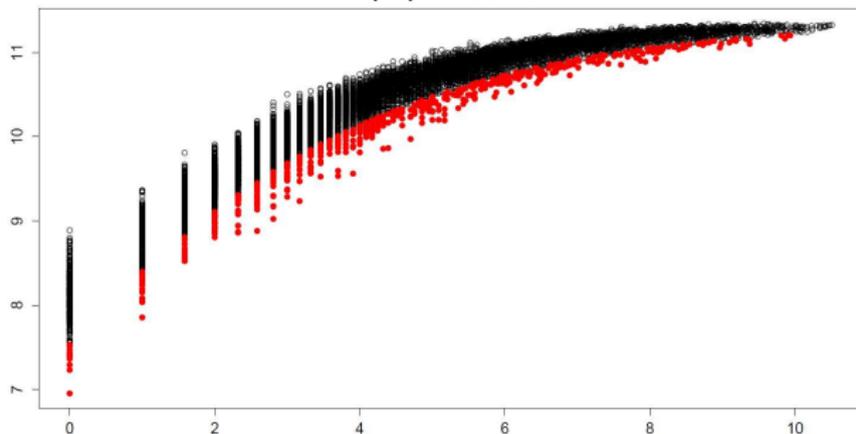
Недостаток: из-за разбиения на интервалы значительная часть узких контекстов пропускается (предвзятый отбор)

V.Dobrynin, D.Patterson, N.Rooney. Contextual document clustering. ECIR, 2004.

Выделение слов, имеющих узкие контексты

Закон Хипса \Rightarrow зависимость $H(w)$ от $\log_2 N_w$ логарифмическая
 Более аккуратный отбор локальных контекстов
 с помощью квантильной регрессии (отсекаем 5% снизу).

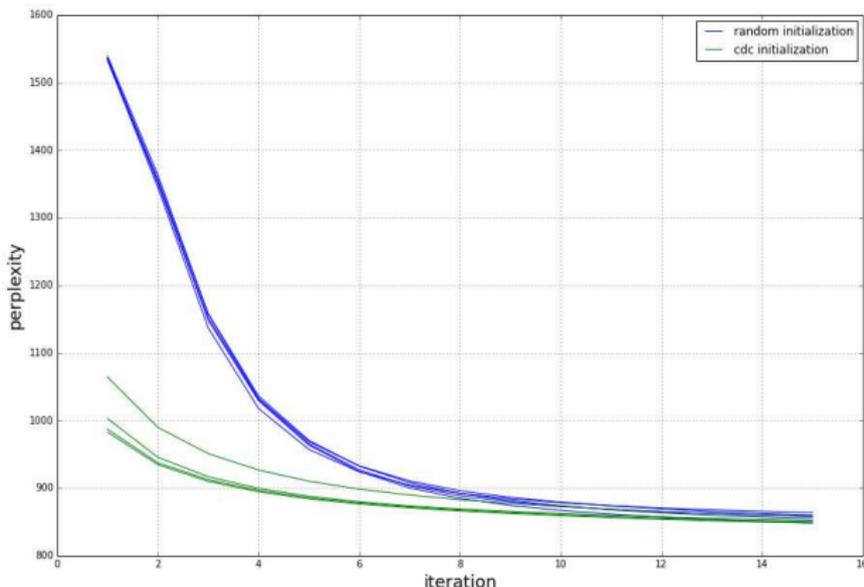
Зависимость энтропии $H(w)$ от документной частоты $\log_2 N_w$



V.Dobrynin, D.Patterson, N.Rooney. Contextual document clustering. ECIR, 2004.
 Алексей Гринчук. Использование контекстной документной кластеризации для
 улучшения качества тематических моделей // ВКР бакалавра, МФТИ. 2015.

Инициализация тематической модели с помощью CDC

Зависимость перплексии от числа итераций (коллекция MMPO)



Алексей Гринчук. Использование контекстной документной кластеризации для улучшения качества тематических моделей // ВКР бакалавра, МФТИ. 2015.

- 1 оптимизировать структуру контекста ($\vec{\gamma}_i$, $\tilde{\gamma}_i$ и др.)
- 2 гипотеза: можно опустить трудоёмкое вычисление N_{tw}
- 3 гипотеза: улучшается интерпретируемость тем
- 4 гипотеза: управление $p(t)$ позволит балансировать темы

Семейство моделей A*RTM: Attentive, Apprehensive, Aware, Adaptive, Automated, Available, Additively Regularized TM

- 1 ARTM: модальности, связи, иерархии, транзакции,...
- 2 формирование «рассказа о себе» для каждой темы:
релевантные фразы, фрагменты, название, суммаризация
- 3 согласование $p(t|w)$ с предобученными эмбедингами LLM
- 4 введение обучаемых параметров в Attentive ARTM
- 5 проверка налету стат-гипотез о согласии распределений
- 6 настройка гиперпараметров в потоке данных (AutoML)

Задача-минимум: научиться решать задачи анализа текстов с использованием тематического моделирования

Задача-максимум: получить новый научный результат

виды деятельности	оценка
теоретическая задача	X
решение прикладной задачи	$10X$
обзор по последним РТМ/NTM	$10X$
участие в проекте	$20X$
работа над открытой проблемой	$25X$

где X — оценка за вид деятельности по 5-балльной шкале.
score — суммарная оценка по всем видам деятельности.

Итоговая оценка: $\min(5, \lfloor \text{score}/20 \rfloor)$ по 5-балльной шкале.

Задания к лекции 1

Упражнения на принцип максимума правдоподобия:

1. Биграммная модель коллекции: $p(w|v) = \xi_{wv}$,

где v — слово, идущее в тексте перед w .

Найти параметры модели ξ_{wv} .

2. Биграммная модель документов: $p(w|v, d) = \xi_{dvw}$.

Найти параметры модели ξ_{dvw} .

Подсказка: применить условия ККТ или основную лемму.

3. Творческое задание (возможны разные решения)

Предложите модель, разделяющую роли слов в текстах:

— тематические слова

— специфичные слова документа (шум)

— слова общей лексики (фон)

Подсказка 1: искать распределение ролей слов $p(r|w)$, $r \in \{\text{т, ш, ф}\}$.

Подсказка 2: можно разреживать $p(r|w)$ для жёсткого определения ролей.

Подсказка 3: можно использовать документную частоту слов.

4. Пользуясь основной леммой, докажите, что регуляризатор битермов эквивалентен добавлению псеводокументов d_u в исходную коллекцию (см. слайд 13)

Прикладная исследовательская задача:

автоматическое выделение научных терминов (АТЕ)

- Дано:
коллекция размеченных текстов конкурса ruTermEval;
неразмеченная коллекция текстов той же тематики
- Найти:
метод АТЕ на основе комбинирования ARTM и TopMine;
обоснование, что синтаксические и др. методы не нужны;
зависимость качества АТЕ от объёма неразмеченных данных
- Критерий:
качество АТЕ (Prec, Rec, F1) на размеченных данных

- 1 Открытые датасеты (английский): 20NG, NIPS, KOS
- 2 Ранжированные результаты поиска научных статей (по данным eLibrary, arXiv, PubMed)
- 3 Научно-популярные статьи: ПостНаука, Элементы, Хабр,...
- 4 Техноблоги: Хабр (русский), TechCrunch (английский)
- 5 Данные социальных сетей: VK, Twitter, Telegram,...
- 6 Статьи по Complexity Sciences (хронокарта науки)
 - Википедия
 - Новостной поток (20 источников на русском языке)
 - Данные кадровых агентств: резюме + вакансии
 - Транзакции клиентов Sberbank DSD 2016
 - Акты арбитражных судов РФ

- «Тематизатор» для социо-гуманитарных исследований:
 - пользователь задаёт грубый фильтр текстового потока;
 - задача: «классифицировать иголки в стоге сена»,
 - разделив темы на информативные и мусорные,
 - выделив аспекты и тональности в каждой теме;
 - конечная цель: $q \& q$ аналитика проблемной среды,
 - реализация данного сценария как модуля в среде Orange
- «Мастерская знаний» для научного поиска:
 - пользователь строит тематические подборки статей,
 - поисковая выдача формируется моделью SciRus;
 - задача: показать пользователю тематику подборки;
 - понадобится: автоматическое выделение терминов,
 - выделение тематических фраз из документов,
 - автоматическое именование и суммаризация тем;
 - конечная цель: помочь в понимании предметной области

- 1 Тематические модели внимания последовательного текста
- 2 Проблема несбалансированности тем в коллекции
- 3 Измерение интерпретируемости тем (когерентность)
- 4 Обеспечение 100%-й интерпретируемости тем
- 5 Автоматическое именованное и суммаризация тем
- 6 Калибровка моделей тематической фильтрации
- 7 Согласование тем с предобученными эмбедингами LLM
- 8 Статистические оценки состоятельности тем
- 9 Обнаружение новых тем или трендов в потоке текстов
- 10 Обеспечение устойчивости и полноты множества тем
- 11 Автоматический подбор гиперпараметров, AutoML
- 12 Гиперграфовые тематические модели для RecSys