# :::::::::::::::::: BigARTM ::::::::::::::::::
# от лего-конструктора тематических моделей к сервисам разведочного поиска

Константин Воронцов
(МФТИ • AITHEA)

DataFest[6] • 10–11 мая 2019

# Содержание

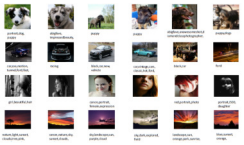
# Тематическое моделирование и его приложения

Тематическое моделирование — «мягкая кластеризация» коллекции текстов

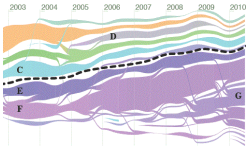
разведочный поиск в электронных библиотеках

персонализированный поиск в соцсетях

мультимодальный поиск текстов и изображений



детектирование и трекинг новостных сюжетов

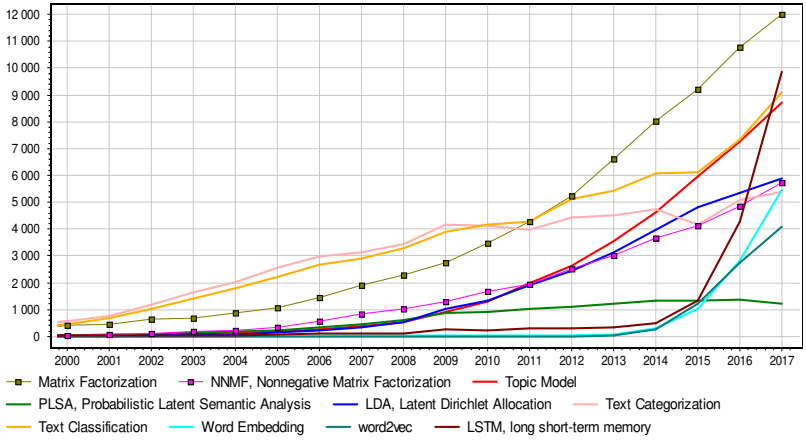



навигация по большим текстовым коллекциям


управлением диалогом в разговорном интеллекте

# Тематическое моделирование и смежные области исследований

Динамика цитирования в академических публикациях, по данным Google Scholar:

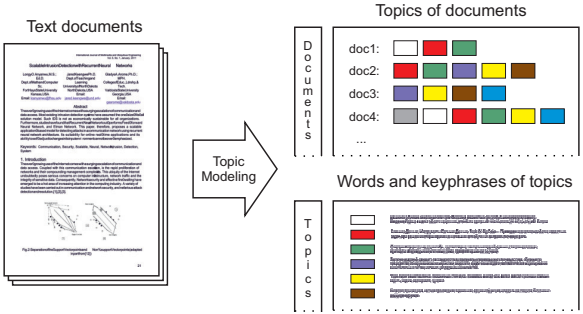


# Тематическое моделирование коллекции текстовых документов

**Дано:** $p(w|d) = \frac{n_{dw}}{n_d}$ — частотное распределение слов $w$ в документах $d$

**Найти:** $p(t|d) = \theta_{td}$ — матрица $\Theta$ распределений тем $t$ в документах $d$

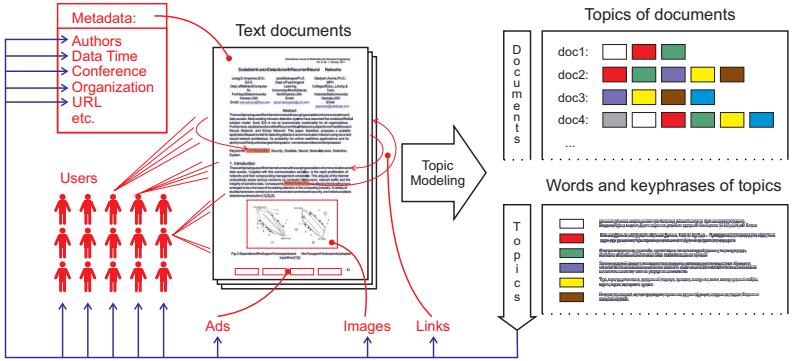
$p(w|t) = \phi_{wt}$ — матрица $\Phi$ распределений слов $w$ в темах $t$




*Hofmann T.* Probabilistic latent semantic indexing. SIGIR 1999.

# Тематическое моделирование мультимодальных данных

Темы определяют распределения термов различных *модальностей* $p(w|t)$:
$p(\text{слово}|t)$, $p(n\text{-грамма}|t)$, $p(\text{автор}|t)$, $p(\text{время}|t)$, $p(\text{категория}|t)$, $p(\text{источник}|t)$,
$p(\text{тег}|t)$, $p(\text{ссылка}|t)$, $p(\text{баннер}|t)$, $p(\text{геолокация}|t)$, $p(\text{пользователь}|t), \ldots$



# Тематическое моделирование — это задача матричного разложения

**Дано:** $p(w|d) = \frac{n_{dw}}{n_d}$ — частотное распределение термов $w$ в документах $d$

**Найти:** $p(t|d) = \theta_{td}$ — матрица $\Theta$ распределений тем $t$ в документах $d$

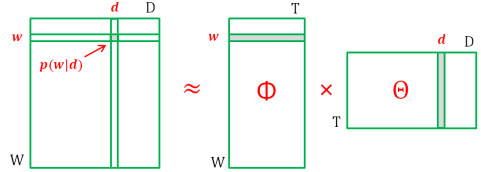
$p(w|t) = \phi_{wt}$ — матрица $\Phi$ распределений термов $w$ в темах $t$

**Критерий:** максимум log-правдоподобия тематической модели

$$\sum_{d\in D}\sum_{w\in d} n_{dw} \ln \sum_{t\in T} \phi_{wt}\theta_{td} \to \max_{\Phi,\Theta}$$

при ограничениях $\phi_{wt} \geqslant 0$, $\theta_{td} \geqslant 0$, $\sum_w \phi_{wt} = 1$, $\sum_t \theta_{td} = 1$.

Это задача стохастического матричного разложения, $T$ — заданное число тем:



# ARTM: тематическая модель с аддитивной регуляризацией и модальностями

Максимизация суммы log-правдоподобий модальностей $W_m$ и регуляризаторов $R_i$:

$$\sum_{m\in M}\tau_m \sum_{d\in D}\sum_{w\in W_m} n_{dw} \ln \sum_{t\in T}\phi_{wt}\theta_{td} + \sum_i \tau_i R_i(\Phi,\Theta) \;\to\; \max_{\Phi,\Theta}$$

EM-алгоритм: метод простой итерации для решения системы уравнений

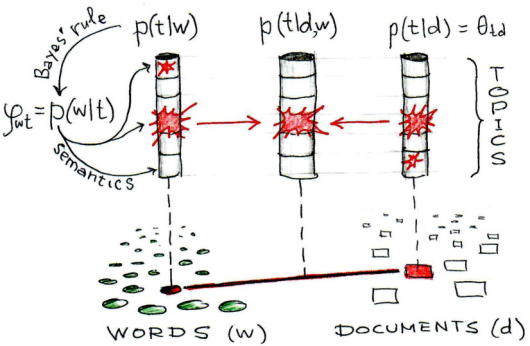
$$\text{E-шаг:}\quad p_{tdw} \equiv p(t|d,w) = \operatorname*{norm}_{t\in T}\left(\phi_{wt}\theta_{td}\right)$$

$$\text{M-шаг:}\quad \begin{cases} \phi_{wt} = \operatorname*{norm}_{w\in W_m}\left(\sum_{d\in D}\tau_m n_{dw} p_{tdw} + \sum_i \tau_i \phi_{wt}\frac{\partial R}{\partial \phi_{wt}}\right) \\ \theta_{td} = \operatorname*{norm}_{t\in T}\left(\sum_{w\in d}\tau_m n_{dw} p_{tdw} + \sum_i \tau_i \theta_{td}\frac{\partial R}{\partial \theta_{td}}\right) \end{cases}$$

*K.Vorontsov, O.Frei, M.Apishev et al.* Non-bayesian additive regularization for multimodal topic modeling of large collections. CIKM TM workshop, 2015.

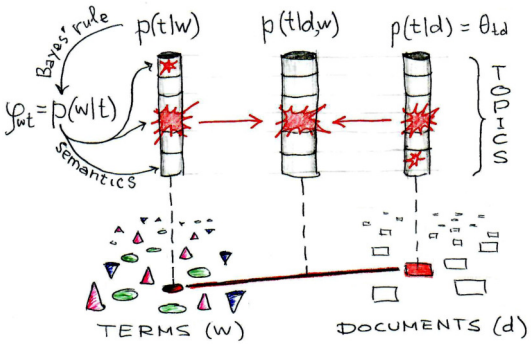
# Тематические эмбединги (векторные представления) слов и документов

- Коллекция текстов — это двудольный граф с рёбрами $(d, w)$
- Эмбединги документов $p(t|d)$, термов $p(t|w)$ и термов-в-контексте $p(t|d, w)$
- Темы интерпретируются благодаря распределению слов $p(w|t) = p(t|w)\frac{p(w)}{p(t)}$



# Мультимодальные тематические эмбединги (векторные представления)

- Каждый терм каждой модальности получает тематический эмбединг
- Каждая тема описывается распределением термов по каждой из модальностей
- Через темы смыслы слов передаются термам всех остальных модальностей



# BigARTM: библиотека тематического моделирования

**Ключевые возможности:**

- Онлайновый параллельный мультимодальный регуляризованный EM-алгоритм
- Пакетная обработка больших данных — коллекция не хранится в памяти
- Встроенная библиотека регуляризаторов и мер качества

**Сообщество:**

- Открытый код https://github.com/bigartm
  (discussion group, issue tracker, pull requests)
- Документация http://bigartm.org

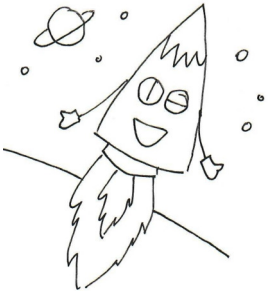
**Лицензия и среда разработки:**

- Свободная коммерческая лицензия (BSD 3-Clause)
- Кросс-платформенность: Windows, Linux, MacOS (32/64 bit)
- Интерфейсы API: command-line, C++, and Python

# Качество и скорость: BigARTM vs Gensim и Vowpal Wabbit

3.7M статей Википедии, 100K слов

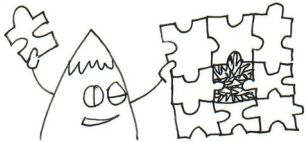
| | | $T = 50$ | | $T = 200$ | |
|---|---|---|---|---|---|
| | проц. | минут | перплексия | минут | перплексия |
| BigARTM | 1 | 42 | 5117 | 83 | 3347 |
| BigARTM async | 1 | 25 | 5131 | 53 | 3362 |
| VowpalWabbit | 1 | 50 | 5413 | 154 | 3960 |
| Gensim | 1 | 142 | 4945 | 637 | 3241 |
| BigARTM | 4 | 12 | 5216 | 26 | 3520 |
| BigARTM async | 4 | 7 | 5353 | 16 | 3634 |
| Gensim | 4 | 88 | 5311 | 315 | 3583 |
| BigARTM | 8 | 8 | 5648 | 15 | 3929 |
| BigARTM async | 8 | 5 | 6220 | 10 | 4309 |
| Gensim | 8 | 88 | 6344 | 288 | 4263 |



*D.Kochedykov, M.Apishev, L.Golitsyn, K.Vorontsov.* Fast and Modular Regularized Topic Modelling. FRUCT ISMW, 2017.

# BigARTM упрощает модульную разработку тематических моделей

Для построения сложных моделей в BigARTM не нужны ни математические выкладки, ни программирование «с нуля».



| Этапы моделирования | Bayesian TM | ARTM | |
|---|---|---|---|
| | Анализ требований | Анализ требований | |
| *Формализация:* | Вероятностная порождающая модель данных | Стандартные критерии | Свои критерии |
| *Алгоритмизация:* | Байесовский вывод для данной порождающей модели (VI, GS, EP) | Общий регуляризованный EM-алгоритм для любых моделей | |
| *Реализация:* | Исследовательский код (Matlab, Python, R) | Промышленный код BigARTM (C++, Python API) | |
| *Оценивание:* | Исследовательские метрики, исследовательский код | Стандартные метрики | Свои метрики |
| | Внедрение | Внедрение | |

-- *нестандартизуемые этапы, уникальная разработка для каждой задачи*

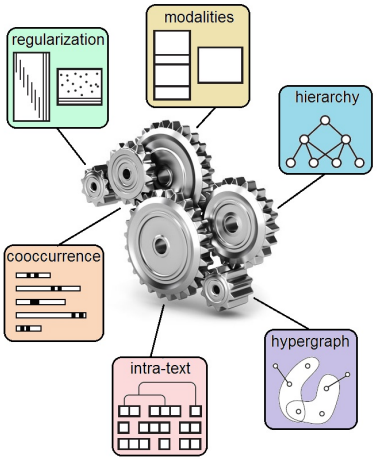
-- *стандартизуемые этапы*

# Шесть ключевых механизмов BigARTM

Благодаря ARTM, эти механизмы можно комбинировать в любых сочетаниях:

1. регуляризация
2. модальности
3. иерархия тем
4. парная встречаемость термов
5. обработка контекстных эмбедингов
6. гиперграфы транзакций

Новые механизмы позволяют учитывать порядок слов в обход гипотезы «мешка слов»



## Механизм иерархического тематического моделирования

**Шаг 1.** Строим модель верхнего уровня с небольшим числом тем.

**Шаг $k$.** Пусть модель с множеством тем $T$ уже построена.
Строим следующий уровень — множество дочерних тем $S$ (subtopics), $|S| > |T|$.

Родительские темы $t$ — *псевдо-документы* с частотами слов $n_{wt} = \phi_{wt} n_t$:

$$\sum_{t \in T} \sum_{w \in W} n_{wt} \ln \sum_{s \in S} \phi_{ws} \theta_{st} \;\to\; \max,$$

где $\theta_{st} = p(s|t)$ — вероятность подтемы $s$ в родительской теме $t$.

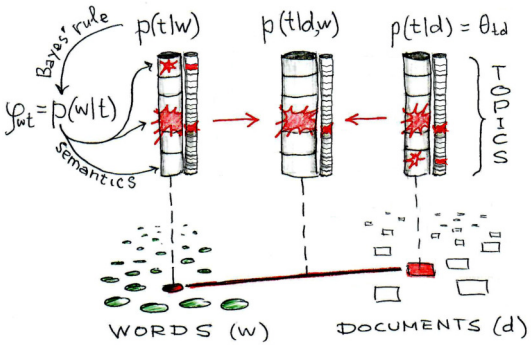
---

*N.A.Chirkova, K.V.Vorontsov.* Additive Regularization for Hierarchical Multimodal Topic Modeling. JMLDA, 2016.

*A.V.Belyy, M.S.Seleznova, A.K.Sholokhov, K.V.Vorontsov.* Quality Evaluation and Improvement for Hierarchical Topic Modeling. Dialogue 2018.

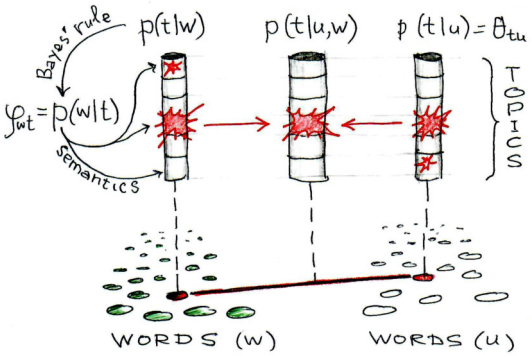
# Механизм иерархического тематического моделирования

- Послойный алгоритм разбиения родительских тем на дочерние подтемы
- На дочернем уровне родительские темы превращаются в *псевдо-документы*
- Связь «много-ко-многим»: дочерняя тема может иметь много родительских



# Механизм тематического моделирования по частотам пар слов

- Дистрибутивная семантика: смысл слова определяется всеми его контекстами
- Слово индуцирует *псевдо-документ* — мешок всех его контекстов
- Тематические векторные представления слов обладают свойствами word2vec

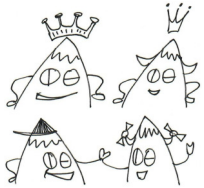


# word2vec и ARTM на задачах аналогии слов

Тематические векторные эмбеддинги объединяют в себе «лучшее от двух миров»:

- **ARTM**: интерпретируемость и разреженность компонент векторов слов
- **word2vec**: интерпретируемость векторных операций над векторами слов

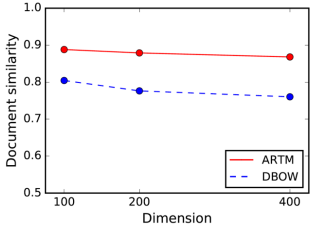
| Операция | Результат ARTM | Результат word2vec |
|---|---|---|
| king – boy + girl | *queen*, princess, lord, prince | *queen*, princess, regnant, kings |
| moscow – russia + spain | *madrid*, barcelona, aires, buenos | *madrid*, barcelona, valladolid, malaga |
| india – russia + ruble | *rupee*, birbhum, pradesh, madhaya | *rupee*, rupiah, devalued, debased |
| cars – car + computer | *computers*, software, servers, implementations | *computers*, software, hardware, microcomputers |



*A.Potapenko, A.Popov, K.Vorontsov.* Interpretable probabilistic embeddings: bridging the gap between topic models and neural networks. AINL-6, 2017.

## word2vec и ARTM в задаче семантической близости документов

**ArXiv triplets dataset:** 20K троек статей:
⟨статья A, схожая статья B, непохожая статья C⟩




- обучение по 1M текстов статей ArXiv
- тестирование на триплетах ArXiv
- конкурент DBOW: paragraph2vec [Dai et. al, 2015]

ARTM превосходит модель DBOW (distributed bag-of-words).

---

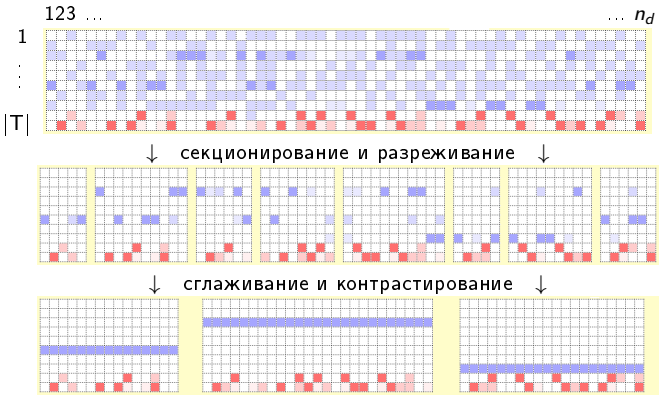
*Andrew Dai, Cristopher Olah, Quoc Le.* Document Embedding with Paragraph Vectors, CoRR, 2015

*A.Potapenko, A.Popov, K.Vorontsov.* Interpretable probabilistic embeddings: bridging the gap between topic models and neural networks. AINL-6, 2017.

# Обработка контекстных эмбедингов на примере тематической сегментации

Документ $d = \{w_1, \ldots, w_{n_d}\}$, $n_d$ — длина документа $d$

*Контекстные эмбединги* — матрица вероятностей тем $p(t|d, w_i)$ размера $T \times n_d$




# Механизм обработки контекстных эмбедингов — регуляризация E-шага

$\Pi = \big(p_{tdw} = p(t|d,w)\big)_{T\times D\times W}$ — трёхмерная матрица контекстных эмбедингов

Максимизация log правдоподобия с регуляризаторами $R$ и $\tilde{R}$:

$$\sum_{d\in D}\sum_{w\in d} n_{dw} \ln \sum_{t\in T} \phi_{wt}\theta_{td} + R(\Pi(\Phi,\Theta)) + \tilde{R}(\Phi,\Theta) \;\to\; \max_{\Phi,\Theta}.$$

EM-алгоритм: метод простой итерации для системы уравнений

E-шаг:
$$\begin{cases} p_{tdw} = \underset{t\in T}{\text{norm}}\big(\phi_{wt}\theta_{td}\big) \\ \tilde{p}_{tdw} = p_{tdw}\left(1 + \frac{1}{n_{dw}}\left(\frac{\partial R(\Pi)}{\partial p_{tdw}} - \sum_{z\in T} p_{zdw}\frac{\partial R(\Pi)}{\partial p_{zdw}}\right)\right) \end{cases}$$

M-шаг:
$$\begin{cases} \phi_{wt} = \underset{w\in W}{\text{norm}}\left(\sum_{d\in D} n_{dw}\tilde{p}_{tdw} + \phi_{wt}\frac{\partial \tilde{R}}{\partial \phi_{wt}}\right) \\ \theta_{td} = \underset{t\in T}{\text{norm}}\left(\sum_{w\in d} n_{dw}\tilde{p}_{tdw} + \theta_{td}\frac{\partial \tilde{R}}{\partial \theta_{td}}\right) \end{cases}$$

# Механизм тематического моделирования транзакционных данных

Выборка может содержать не только пары $(d, w)$, но также тройки, четвёрки, ..., $n$-ки элементов разных модальностей.

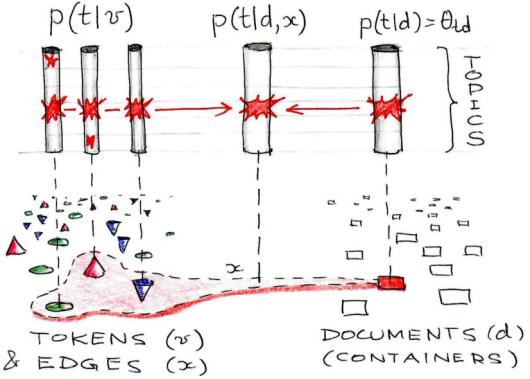
**Примеры:**

- **Данные социальной сети:**
  $(d, u, w)$ — пользователь $u$ записал слово $w$ в блоге $d$
- **Данные сети интернет-рекламы:**
  $(u, d, b)$ — пользователь $u$ кликнул баннер $b$ на странице $d$
- **Данные финансовых организаций:**
  $(b, s, g)$ — покупатель $u$ купил у продавца $s$ товар $g$

**Задача:** по выборке рёбер гиперграфа выявить латентные темы его вершин.

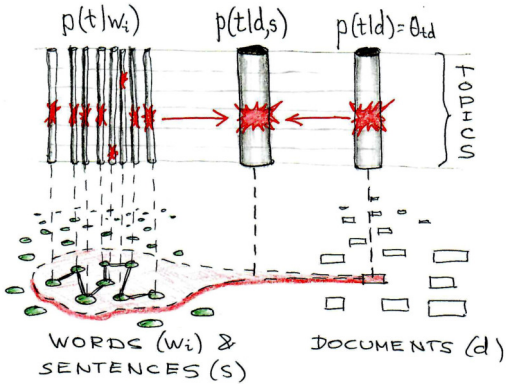
# Интерпретируемые эмбедингы транзакционных данных

- *Гиперграф* — это система подмножеств вершин-термов
- Транзакция = подмножество термов = ребро гиперграфа
- Транзакция тем более вероятна, чем больше общих тем имеют её термы



# Интерпретируемые эмбединги предложений

- Предложение — это наиболее семантически однородная единица языка
- Предложение = подмножество слов = ребро гиперграфа
- Предложение тем более вероятно, чем больше общих тем имеют его слова



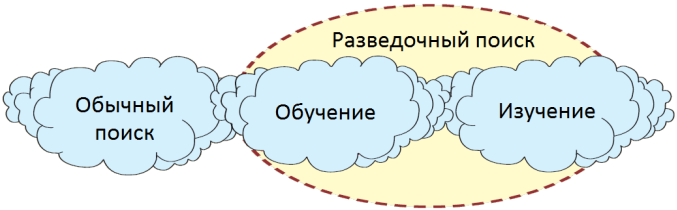
# Гиперграфовые тематические модели языка

Ребром гиперграфа можно описать любое подмножество термов, связанных друг с другом по смыслу и порождаемых одной общей темой:

- предложение
- синтагма — поддерево синтаксического дерева
- именная группа
- факт «объект, субъект, действие»
- связанные термы соседних предложений: синонимы, гиперонимы, холонимы
- лексическая цепочка
- текст сообщения и его автор
- финансовая транзакция с текстом платёжного поручения

# Концепция разведочного информационного поиска (exploratory search)

- пользователь может не знать ключевых терминов предметной области
- запросом может быть текст произвольной длины или даже подборка текстов
- информационная потребность пользователя — систематизация знаний



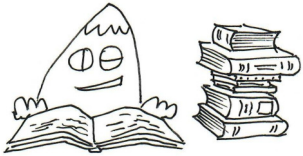

*Gary Marchionini*. Exploratory Search: from finding to understanding. 2006.

## Поиск тематически близких документов

$\theta_{tq} = p(t|q)$ — тематический профиль текста запроса $q$
$\theta_{td} = p(t|d)$ — тематические профили документов $d$ из коллекции

Косинусная мера близости документа $d$ и запроса $q$:

$$\text{sim}(q,d) = \frac{\sum_t \theta_{tq}\theta_{td}}{\left(\sum_t \theta_{tq}^2\right)^{1/2}\left(\sum_t \theta_{td}^2\right)^{1/2}}.$$



Ранжируем документы $d \in D$ по убыванию $\text{sim}(q,d)$
Выдача тематического поиска — $k$ первых документов.

Реализация: *векторный поиск*, либо *инвертированный индекс* для быстрого поиска документов $d$ по каждой из тем $t$ запроса

*A.Ianina, K.Vorontsov.* Multi-objective topic modeling for exploratory search in tech news. AINL, 2017.

# Две коллекции новостей про технологии

## Habrahabr.ru

175 143 статей на русском языке

**Шесть модальностей в текстах:**

- 10 552 слов (униграмм)
- 742 000 биграмм
- 524 авторов статей
- 10 000 авторов комментариев
- 2546 тегов
- 123 хаба

## TechCrunch.com

759 324 статей на английском языке

**Четыре модальности в текстах:**

- 11 523 слов (униграмм)
- 1.2 млн. биграмм
- 605 авторов
- 184 категорий

# Методика оценивания качества разведочного поиска

**Поисковый запрос**

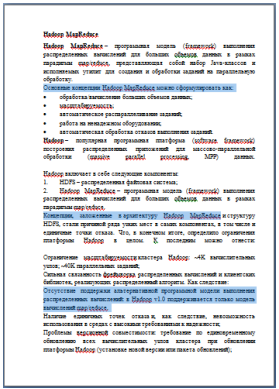
ключевые слова или фрагменты текста, одна страница A4

**Поисковая выдача**

документы, тематически близкие к документу-запросу

**Два задания асессорам**

- найти как можно больше статей, пользуясь любыми средствами поиска (и засечь время)
- оценить релевантность поисковой выдачи на том же запросе



Пример запроса для разведочного поиска

# Пример: фрагмент запроса «Система IBM Watson»

IBM Watson — суперкомпьютер фирмы IBM, оснащённый вопросно-ответной системой искусственного интеллекта, созданный группой исследователей под руководством Дэвида Феруччи. Его создание — часть проекта DeepQA. Основная задача Уотсона — понимать вопросы, сформулированные на естественном языке, и находить на них ответы в базе данных. Назван в честь основателя IBM Томаса Уотсона.

IBM Watson представляет собой когнитивную систему, которая способна понимать, делать выводы и обучаться. Она также позволяет преобразовывать целые отрасли, различные направления науки и техники. Например, предсказывать появление эпидемий или возникновения очагов природных катастроф в различных регионах, вести мониторинг состояния атмосферы больших городов, оптимизировать бизнес-процессы, узнавать, какие товары будут в тренде в ближайшее время.
... ... ...

**Релевантные тексты:** примеры сервисов и приложений, основа которых — когнитивная платформа IBM Watson, используемые в IBM Watson технологии, вопрос-ответные системы, сопоставление IBM Watson с Wolfram-Alpha.

**Нерелевантные тексты:** общие вопросы искусственного интеллекта, другие коммерческие решения на рынке бизнес-аналитики.

## Тематика запросов разведочного поиска

Примеры заголовков разведочных запросов к Хабру
(объём каждого запроса — около одной страницы A4):

| | |
|---|---|
| Алгоритмы раскраски графов | Система IBM Watson |
| Рекомендательная система Netflix | 3D-принтеры |
| Методики быстрого набора текста | CERN-кластер |
| Космические проекты Илона Маска | AB-тестирование |
| Технологии Hadoop MapReduce | Облачные сервисы |
| Беспилотный автомобиль Google car | Контекстная реклама |
| Криптосистемы с открытым ключом | Марсоход Curiosity |
| Обзор платформ онлайн-курсов | Видеокарты NVIDIA |
| Data Science Meetups в Москве | Распознавание образов |
| Образовательные проекты mail.ru | Сервисы Google scholar |
| Межпланетная станция New horizons | MIT MediaLab Research |
| Языковая модель word2vec | Платформа Microsoft Azure |

# Оценивание качества поиска

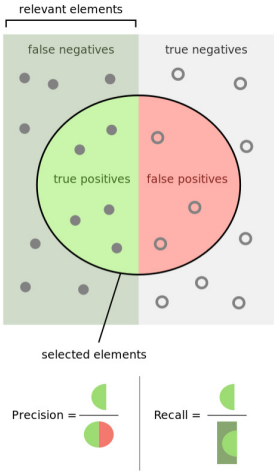
Precision — доля релевантных среди найденных
Recall — доля найденных среди релевантных

$$P = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \text{ — точность (precision)}$$

$$R = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \text{ — полнота, (recall)}$$

$$F_1 = \frac{P + R}{2PR} \text{ — F1-мера}$$

TP (true positive) — найденные релевантные
FP (false positive) — найденные нерелевантные
FN (false negative) — ненайденные релевантные



## Какие модели поиска сравнивались

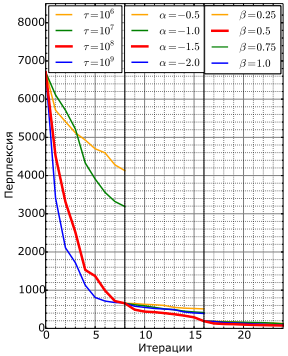
- **assessors:** результаты поиска, выполненного людьми (асессорами)
- **TF-IDF, BM25:** сравнение документов по векторам частот слов
- **word2vec:** нетематические векторные представления слов
- **PLSA:** Probabilistic Latent Semantic Analysis [T.Hofmann, 1999]
- **LDA:** Latent Dirichlet Allocation [D.Blei, A.Ng, M.Jordan, 2003]
- **ARTM:** тематическая модель с тремя регуляризаторами
- **hARTM:** двухуровневая иерархическая тематическая модель

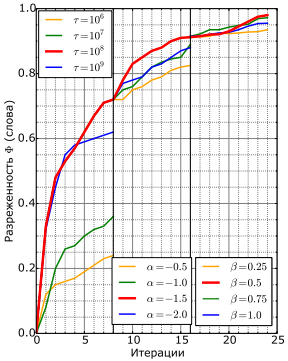
**Дополнительные критерии (регуляризаторы) в ARTM и hARTM:**

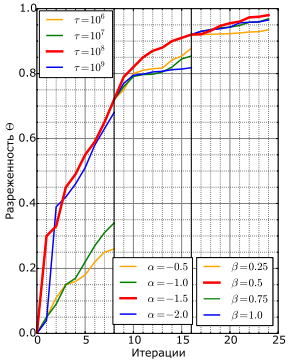
- сделать темы как можно более различными
- сделать профили $p(t|d)$ как можно более разреженными
- сужать область поиска с помощью иерархических профилей $p(t|d)$

# Последовательный подбор коэффициентов регуляризации в моделях ARTM

- декоррелирование распределений терминов в темах ($\tau$),
- разреживание распределений тем в документах ($\alpha$),
- сглаживание распределений терминов в темах ($\beta$).

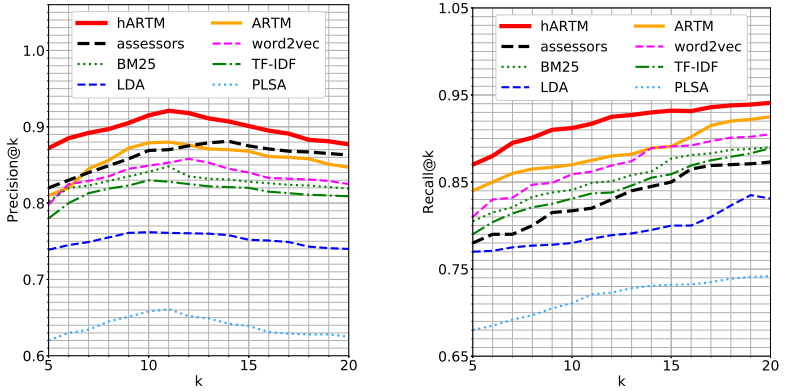


# Сравнение качества поиска с асессорами и простыми моделями

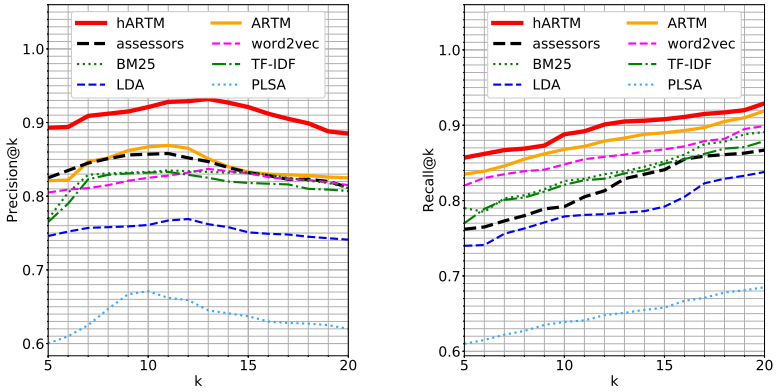
**Habrahabr.** Точность и полнота по первым $k$ позициям поисковой выдачи




*A.Ianina, K.Vorontsov.* Multi-objective topic modeling for exploratory search in tech news. AINL, 2017.

# Сравнение качества поиска с асессорами и простыми моделями

**TechCrunch.** Точность и полнота по первым $k$ позициям поисковой выдачи

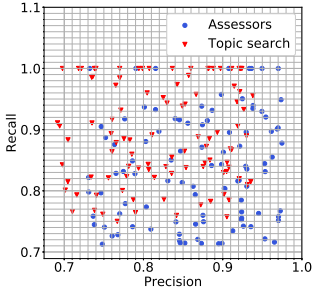



*A.Ianina, K.Vorontsov.* Multi-objective topic modeling for exploratory search in tech news. AINL, 2017.

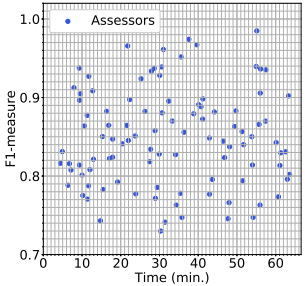
# Результаты измерения точности и полноты по запросам

**Habrahabr**. Качество и время поиска. 100 запросов, 3 асессора на запрос

точность и полнота поиска




время и $F_1$-мера (асессоры)




- среднее время обработки запроса асессором — 30 минут
- точность выше у асессоров, полнота — у поисковика

# Влияние числа тем на качество поиска

Все регуляризаторы и модальности, плоская модель

| | Habrahabr | | | | | | | | TechCrunch | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | acecc | 100 | 150 | **200** | 250 | 300 | 400 | 500 | acecc | 350 | 400 | 450 | **475** | 500 |
| Pr@5 | 0.821 | 0.662 | 0.721 | **0.810** | 0.761 | 0.712 | 0.693 | 0.642 | 0.822 | 0.653 | 0.725 | 0.752 | **0.819** | 0.777 |
| Pr@10 | 0.869 | 0.761 | 0.812 | **0.879** | 0.825 | 0.722 | 0.673 | 0.662 | 0.851 | 0.663 | 0.732 | 0.762 | **0.867** | 0.811 |
| Pr@15 | 0.875 | 0.733 | 0.795 | **0.868** | 0.791 | 0.671 | 0.651 | 0.631 | 0.835 | 0.682 | 0.743 | 0.787 | **0.833** | 0.793 |
| Pr@20 | 0.863 | 0.724 | 0.795 | **0.847** | 0.792 | 0.673 | 0.642 | 0.622 | 0.813 | 0.650 | 0.743 | 0.773 | **0.825** | 0.793 |
| R@5 | 0.780 | 0.732 | 0.807 | **0.840** | 0.821 | 0.805 | 0.721 | 0.671 | 0.762 | 0.731 | 0.762 | 0.793 | **0.835** | 0.817 |
| R@10 | 0.817 | 0.771 | 0.843 | **0.870** | 0.851 | 0.812 | 0.751 | 0.712 | 0.792 | 0.763 | 0.793 | 0.812 | **0.868** | 0.855 |
| R@15 | 0.850 | 0.824 | **0.895** | 0.891 | 0.871 | 0.822 | 0.773 | 0.753 | 0.835 | 0.782 | 0.807 | 0.855 | **0.890** | 0.882 |
| R@20 | 0.873 | 0.857 | 0.905 | **0.925** | 0.892 | 0.855 | 0.771 | 0.764 | 0.867 | 0.792 | 0.823 | 0.862 | **0.919** | 0.903 |

- существует оптимальное число тем
- чем больше коллекция, тем больше оптимум числа тем

## Влияние числа тем на качество поиска

**Habrahabr**. Все регуляризаторы и модальности, три уровня иерархии

| $\lvert T_1\rvert$ | **20** | | **25** | | | | | | | **30** | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $\lvert T_2\rvert$ | 150 | 200 | **250** | | **275** | | | **300** | | 400 | 450 |
| $\lvert T_3\rvert$ | 750 | 800 | 1200 | 1300 | 1300 | **1400** | 1500 | 1500 | 1600 | 3000 | 3500 |
| Pr@5 | 0.625 | 0.743 | 0.840 | 0.852 | 0.869 | **0.872** | 0.870 | 0.805 | 0.771 | 0.705 | 0.672 |
| Pr@10 | 0.648 | 0.754 | 0.851 | 0.867 | 0.882 | **0.915** | 0.901 | 0.811 | 0.799 | 0.722 | 0.694 |
| Pr@15 | 0.632 | 0.752 | 0.850 | 0.872 | 0.878 | **0.895** | 0.889 | 0.809 | 0.785 | 0.729 | 0.703 |
| Pr@20 | 0.629 | 0.745 | 0.845 | 0.861 | 0.871 | 0.877 | **0.882** | 0.803 | 0.778 | 0.710 | 0.681 |
| R@5 | 0.632 | 0.780 | 0.845 | 0.869 | 0.883 | **0.889** | 0.872 | 0.851 | 0.841 | 0.721 | 0.695 |
| R@10 | 0.654 | 0.792 | 0.859 | 0.873 | 0.905 | **0.922** | 0.881 | 0.873 | 0.850 | 0.749 | 0.703 |
| R@15 | 0.675 | 0.805 | 0.874 | 0.892 | 0.932 | **0.942** | 0.905 | 0.889 | 0.863 | 0.787 | 0.725 |
| R@20 | 0.684 | 0.824 | 0.889 | 0.901 | 0.958 | **0.961** | 0.912 | 0.904 | 0.878 | 0.805 | 0.734 |

- существует оптимальное число тем на каждом уровне
- три уровня слегка лучше, чем два; оптимальное число тем увеличивается

# Влияние числа тем на качество поиска

**Habrahabr**. Все регуляризаторы и модальности, два уровня иерархии

| $\|T_1\|$ | **20** | | **25** | | | | | **30** | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $\|T_2\|$ | 150 | 200 | **250** | | **275** | | | **300** | | 400 | 450 |
| Pr@5 | 0.621 | 0.742 | 0.839 | 0.850 | 0.865 | **0.869** | **0.869** | 0.803 | 0.769 | 0.701 | 0.670 |
| Pr@10 | 0.645 | 0.749 | 0.850 | 0.861 | 0.879 | **0.911** | 0.895 | 0.809 | 0.796 | 0.719 | 0.689 |
| Pr@15 | 0.635 | 0.751 | 0.848 | 0.869 | 0.873 | **0.893** | 0.887 | 0.807 | 0.781 | 0.721 | 0.701 |
| Pr@20 | 0.630 | 0.745 | 0.841 | 0.855 | 0.864 | 0.874 | **0.875** | 0.800 | 0.775 | 0.709 | 0.675 |
| R@5 | 0.628 | 0.773 | 0.843 | 0.865 | 0.881 | **0.881** | 0.868 | 0.849 | 0.839 | 0.715 | 0.691 |
| R@10 | 0.652 | 0.782 | 0.855 | 0.871 | 0.902 | **0.918** | 0.877 | 0.871 | 0.845 | 0.745 | 0.699 |
| R@15 | 0.671 | 0.801 | 0.870 | 0.889 | 0.929 | **0.939** | 0.901 | 0.883 | 0.861 | 0.781 | 0.722 |
| R@20 | 0.680 | 0.819 | 0.886 | 0.892 | **0.955** | **0.955** | 0.907 | 0.901 | 0.872 | 0.801 | 0.729 |

- существует оптимальное число тем на каждом уровне
- два уровня лучше, чем один; при этом оптимальное число тем увеличивается

## Влияние числа тем на качество поиска

**TechCrunch**. Все регуляризаторы и модальности, три уровня иерархии

| $\|T_1\|$ | **80** | | **100** | | | | | | | **120** | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $\|T_2\|$ | 300 | 350 | **500** | | **550** | | | **600** | | 700 | 750 |
| $\|T_3\|$ | 1500 | 1700 | 2500 | 2600 | 2600 | **2800** | 3000 | 3000 | 3200 | 4500 | 4700 |
| Pr@5 | 0.655 | 0.707 | 0.751 | 0.792 | 0.887 | **0.893** | 0.890 | 0.789 | 0.722 | 0.703 | 0.678 |
| Pr@10 | 0.678 | 0.712 | 0.773 | 0.823 | 0.895 | **0.922** | 0.905 | 0.805 | 0.741 | 0.722 | 0.692 |
| Pr@15 | 0.692 | 0.715 | 0.775 | 0.831 | 0.902 | **0.921** | 0.907 | 0.821 | 0.743 | 0.725 | 0.703 |
| Pr@20 | 0.687 | 0.709 | 0.761 | 0.819 | 0.889 | 0.885 | **0.898** | 0.809 | 0.736 | 0.719 | 0.683 |
| R@5 | 0.751 | 0.795 | 0.802 | 0.856 | 0.871 | **0.877** | 0.863 | 0.852 | 0.831 | 0.738 | 0.705 |
| R@10 | 0.767 | 0.812 | 0.825 | 0.875 | 0.892 | **0.908** | 0.879 | 0.871 | 0.842 | 0.751 | 0.711 |
| R@15 | 0.772 | 0.824 | 0.841 | 0.887 | 0.912 | **0.927** | 0.901 | 0.893 | 0.854 | 0.772 | 0.721 |
| R@20 | 0.783 | 0.830 | 0.854 | 0.892 | 0.931 | **0.949** | 0.935 | 0.905 | 0.871 | 0.790 | 0.732 |

- существует оптимальное число тем на каждом уровне
- три уровня слегка лучше, чем два; оптимальное число тем увеличивается

## Влияние числа тем на качество поиска

**TechCrunch**. Все регуляризаторы и модальности, два уровня иерархии

| $\|T_1\|$ | **80** | | **100** | | | | | **120** | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $\|T_2\|$ | 300 | 350 | **500** | | **550** | | | **600** | | 700 | 750 |
| Pr@5 | 0.651 | 0.701 | 0.749 | 0.789 | 0.883 | **0.889** | **0.889** | 0.785 | 0.721 | 0.701 | 0.675 |
| Pr@10 | 0.675 | 0.709 | 0.771 | 0.821 | 0.891 | **0.918** | 0.902 | 0.803 | 0.738 | 0.718 | 0.691 |
| Pr@15 | 0.687 | 0.712 | 0.773 | 0.827 | 0.899 | **0.919** | 0.905 | 0.817 | 0.741 | 0.721 | 0.701 |
| Pr@20 | 0.683 | 0.707 | 0.759 | 0.815 | 0.885 | 0.888 | **0.895** | 0.805 | 0.732 | 0.716 | 0.679 |
| R@5 | 0.749 | 0.791 | 0.801 | 0.854 | 0.868 | **0.875** | 0.861 | 0.849 | 0.829 | 0.731 | 0.701 |
| R@10 | 0.765 | 0.809 | 0.823 | 0.873 | 0.890 | **0.904** | 0.875 | 0.867 | 0.835 | 0.745 | 0.708 |
| R@15 | 0.771 | 0.820 | 0.841 | 0.882 | 0.909 | **0.921** | 0.895 | 0.890 | 0.848 | 0.769 | 0.717 |
| R@20 | 0.778 | 0.825 | 0.851 | 0.887 | 0.928 | **0.942** | 0.929 | 0.901 | 0.869 | 0.785 | 0.728 |

- существует оптимальное число тем на каждом уровне
- два уровня лучше, чем один; при этом оптимальное число тем увеличивается

## Влияние модальностей на качество поиска

Все регуляризаторы, три уровня иерархии, оптимальное число тем

**Модальности:** Words, Bigrams, Authors, Comments, Tags, Hubs, Categories

| | Habrahabr | | | | | | TechCrunch | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | acecc | W | Com | WB | WBTH | **All** | acecc | W | C | WB | WBC | **All** |
| Pr@5 | 0.821 | 0.621 | 0.558 | 0.673 | 0.871 | **0.872** | 0.822 | 0.718 | 0.569 | 0.795 | 0.891 | **0.893** |
| Pr@10 | 0.869 | 0.645 | 0.567 | 0.712 | 0.911 | **0.915** | 0.851 | 0.729 | 0.592 | 0.807 | 0.919 | **0.922** |
| Pr@15 | 0.875 | 0.631 | 0.532 | 0.693 | 0.894 | **0.895** | 0.835 | 0.737 | 0.603 | 0.803 | 0.920 | **0.921** |
| Pr@20 | 0.863 | 0.628 | 0.531 | 0.688 | 0.877 | **0.877** | 0.813 | 0.729 | 0.594 | 0.792 | 0.883 | **0.885** |
| R@5 | 0.780 | 0.725 | 0.645 | 0.797 | 0.888 | **0.889** | 0.762 | 0.754 | 0.659 | 0.775 | 0.874 | **0.877** |
| R@10 | 0.817 | 0.748 | 0.652 | 0.812 | 0.921 | **0.922** | 0.792 | 0.778 | 0.671 | 0.808 | 0.908 | **0.908** |
| R@15 | 0.850 | 0.782 | 0.679 | 0.842 | 0.941 | **0.942** | 0.835 | 0.783 | 0.679 | 0.825 | 0.927 | **0.927** |
| R@20 | 0.873 | 0.789 | 0.672 | 0.852 | 0.960 | **0.961** | 0.867 | 0.785 | 0.711 | 0.837 | 0.949 | **0.949** |

- лучше использовать все модальности
- биграммы и категории выигрывают у асессоров
- авторы и комментаторы наименее важны

## Влияние регуляризаторов на качество поиска

Все модальности, три уровня иерархии, оптимальное число тем

**Регуляризаторы:** Decorrelation, Θ-sparsing, Φ-smoothing, Hierarchy interlevel sparsing

| | Habrahabr | | | | | TechCrunch | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | нет | D | DΘ | DΘΦ | DΘΦH | нет | D | DΘ | DΘΦ | DΘΦH |
| Pr@5 | 0.628 | 0.772 | 0.771 | 0.865 | **0.872** | 0.652 | 0.777 | 0.779 | 0.879 | **0.893** |
| Pr@10 | 0.653 | 0.781 | 0.812 | 0.883 | **0.915** | 0.679 | 0.788 | 0.819 | 0.895 | **0.922** |
| Pr@15 | 0.642 | 0.785 | 0.792 | 0.891 | **0.895** | 0.669 | 0.791 | 0.798 | 0.901 | **0.921** |
| Pr@20 | 0.643 | 0.771 | 0.783 | 0.875 | **0.877** | 0.673 | 0.775 | 0.792 | **0.892** | 0.885 |
| R@5 | 0.692 | 0.820 | 0.805 | 0.875 | **0.889** | 0.673 | 0.825 | 0.812 | 0.869 | **0.877** |
| R@10 | 0.714 | 0.831 | 0.834 | 0.905 | **0.922** | 0.685 | 0.856 | 0.845 | 0.881 | **0.908** |
| R@15 | 0.725 | 0.847 | 0.867 | 0.921 | **0.942** | 0.712 | 0.877 | 0.869 | 0.912 | **0.927** |
| R@20 | 0.735 | 0.873 | 0.891 | 0.943 | **0.961** | 0.723 | 0.892 | 0.895 | 0.934 | **0.949** |

- Лучше использовать все регуляризаторы
- Модели со слабой регуляризацией (PLSA, LDA) не конкурентны

## Влияние функции близости эмбедингов на качество поиска

Все регуляризаторы и модальности, три уровня иерархии, оптимальное число тем
**Функции близости:** Euclidean, Cosine, Manhattan, Hellinger, Kullback–Leibler

| | Habrahabr | | | | | TechCrunch | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Eu | **cos** | Ma | He | KL | Eu | **cos** | Ma | He | KL |
| Pr@5 | 0.652 | **0.872** | 0.772 | 0.725 | 0.741 | 0.647 | **0.893** | 0.752 | 0.742 | 0.735 |
| Pr@10 | 0.693 | **0.915** | 0.798 | 0.749 | 0.772 | 0.658 | **0.922** | 0.794 | 0.758 | 0.751 |
| Pr@15 | 0.695 | **0.895** | 0.803 | 0.737 | 0.751 | 0.672 | **0.921** | 0.801 | 0.745 | 0.742 |
| Pr@20 | 0.671 | **0.877** | 0.789 | 0.731 | 0.738 | 0.652 | **0.885** | 0.793 | 0.739 | 0.738 |
| R@5 | 0.693 | **0.889** | 0.721 | 0.742 | 0.833 | 0.688 | **0.877** | 0.708 | 0.733 | 0.858 |
| R@10 | 0.715 | **0.922** | 0.732 | 0.775 | 0.868 | 0.692 | **0.908** | 0.715 | 0.753 | 0.872 |
| R@15 | 0.732 | **0.942** | 0.739 | 0.791 | 0.892 | 0.724 | **0.927** | 0.719 | 0.785 | 0.895 |
| R@20 | 0.741 | **0.961** | 0.721 | 0.812 | 0.902 | 0.732 | **0.949** | 0.711 | 0.808 | 0.901 |

- косинусная функция близости уверенно лидирует

# Направления дальнейшего развития BigARTM и его приложений

**BigARTM:**

- удобство построения тематических моделей последовательного текста
- эффективная встроенная тематическая сегментация
- автоматический подбор коэффициентов регуляризации и весов модальностей
- автоматическое именование и суммаризация тем
- интерпретируемые, разреженные, иерархические, мультиязычные тематические эмбедингы, предобученные по Википедии

**Приложения** — поисково-рекомендательные сервисы:

- создание тематических подборок и «карт знаний»
- составление рефератов и дайджестов
- тематический анализ новостных потоков

# Литература


- *К.В.Воронцов*. Обзор вероятностных тематических моделей. 2018. – NEW!
  http://www.MachineLearning.ru/wiki/images/d/d5/Voron17survey-artm.pdf
- *К.В.Воронцов*. Аддитивная регуляризация тематических моделей коллекций текстовых документов. Доклады РАН. 2014.
- *K.Vorontsov, A.Potapenko*. Additive regularization of topic models. Machine Learning, 2015.
- *O.Frei, M.Apishev*. Parallel non-blocking deterministic algorithm for online topic modeling. AIST 2016.
- *N.Chirkova, K.Vorontsov*. Additive regularization for hierarchical multimodal topic modeling. JMLDA, 2016.
- *A.Ianina, K.Vorontsov*. Multi-objective topic modeling for exploratory search in tech news. AINL, 2017.
- *A.Potapenko, A.Popov, K.Vorontsov*. Interpretable probabilistic embeddings: bridging the gap between topic models and neural networks. AINL, 2017.
- *V.Alekseev, V.Bulatov, K.Vorontsov*. Intra-Text Coherence as a Measure of Topic Models Interpretability. Dialogue, 2018.
- *A.Belyy, M.Seleznova, A.Sholokhov, K.Vorontsov*. Quality Evaluation and Improvement for Hierarchical Topic Modeling. Dialogue, 2018.
- *N.Skachkov, K.Vorontsov*. Improving topic models with segmental structure of texts. Dialogue, 2018.