Построение регуляризованных тематических моделей в BigARTM

Мурат Апишев
great-mel@yandex.ru
MelLain@github.com

МГУ им. М. В. Ломоносова, Яндекс, ШАД

17 марта, 2017

Содержание занятия

1. **Теоретическое напоминание**
   - Аддитивная регуляризация ТМ
   - Мультимодальные тематические модели
   - Краткий обзор библиотеки BigARTM
2. **Эксперименты в BigARTM**
   - Стратегии регуляризации
   - Общие рекомендации по подбору параметров
   - Практические советы и оценивание моделей
3. **Реальный эксперимент**
   - Подготовка эксперимента
   - Проведение эксперимента
   - Оценивание результатов

Тематическое моделирование

Тематическое моделирование (*topic modeling*) — статистический анализ текстов для выявления латентных тем в коллекциях документов.

Тема — терминология предметной области, набор терминов (слов или n-грамм), часто со-встречающихся в документах.

Вероятностная тематическая модель:

- *тема* t — распределение $p(w|t)$ над терминами w
- *документ* d — распределение $p(t|d)$ над темами t

Мешок слов

Мешок слов (Bag-Of-Words) — представление текстовых данных, в котором учитывается только частота встречаемости слов в документах. Порядок слов игнорируется.

Исходное предложение: `I can drink a milk can`

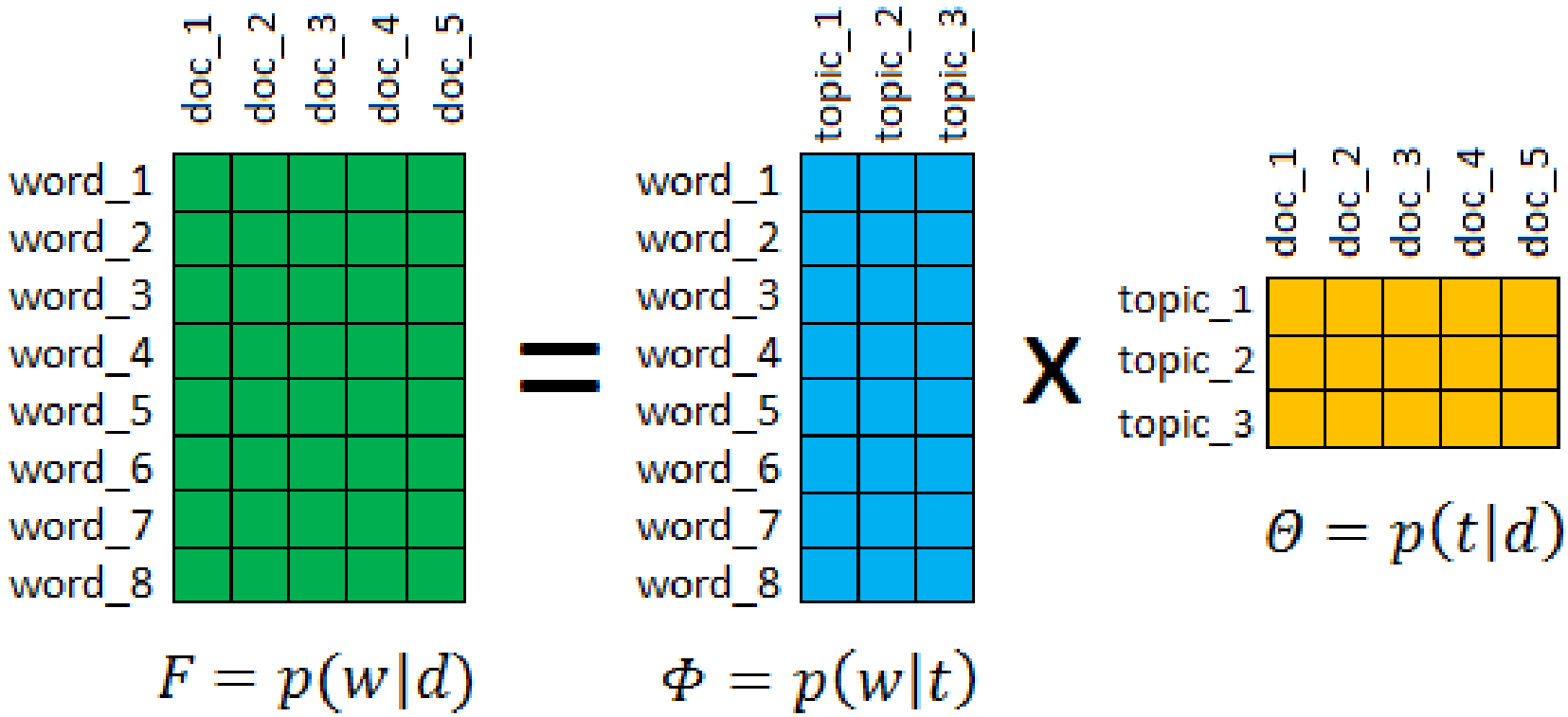
Его мешок слов:

```
I: 1
can: 2
drink: 1
a: 1
milk: 1
```

Проще, но теряется много полезной информации.

Матричное разложение

Если представить данные в виде матрицы $\|p(w|d)\|$,
то тематическая модель — это *матричное разложение*:



Постановка задачи ARTM и регуляризованный EM-алгоритм

Максимизация логарифма правдоподобия с регуляризатором:

$$\sum_{d\in D}\sum_{w\in d} n_{dw} \ln \sum_t \phi_{wt}\theta_{td} + R(\Phi,\Theta) \to \max_{\Phi,\Theta}$$

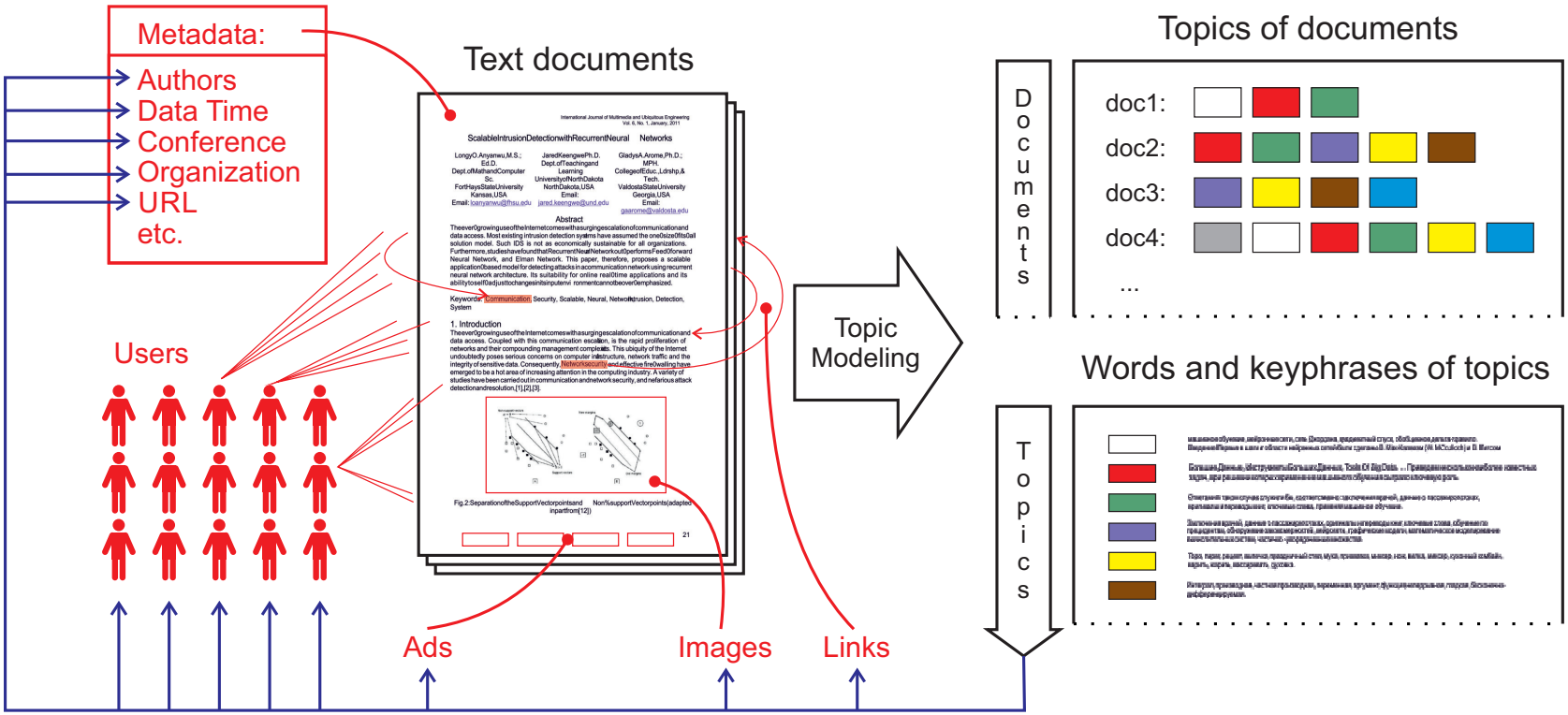
EM-алгоритм: метод простых итераций для системы уравнений

$$\text{E-шаг:} \quad p_{tdw} = \underset{t\in T}{\text{norm}}\left(\phi_{wt}\theta_{td}\right)$$

$$\text{M-шаг:} \quad \begin{cases} \phi_{wt} = \underset{w\in W}{\text{norm}}\left(n_{wt} + \phi_{wt}\dfrac{\partial R}{\partial \phi_{wt}} \right), & n_{wt} = \sum\limits_{d\in D} n_{dw} p_{tdw} \\ \theta_{td} = \underset{t\in T}{\text{norm}}\left(n_{td} + \theta_{td}\dfrac{\partial R}{\partial \theta_{td}} \right), & n_{td} = \sum\limits_{w\in d} n_{dw} p_{tdw} \end{cases}$$

Мультимодальная тематическая модель

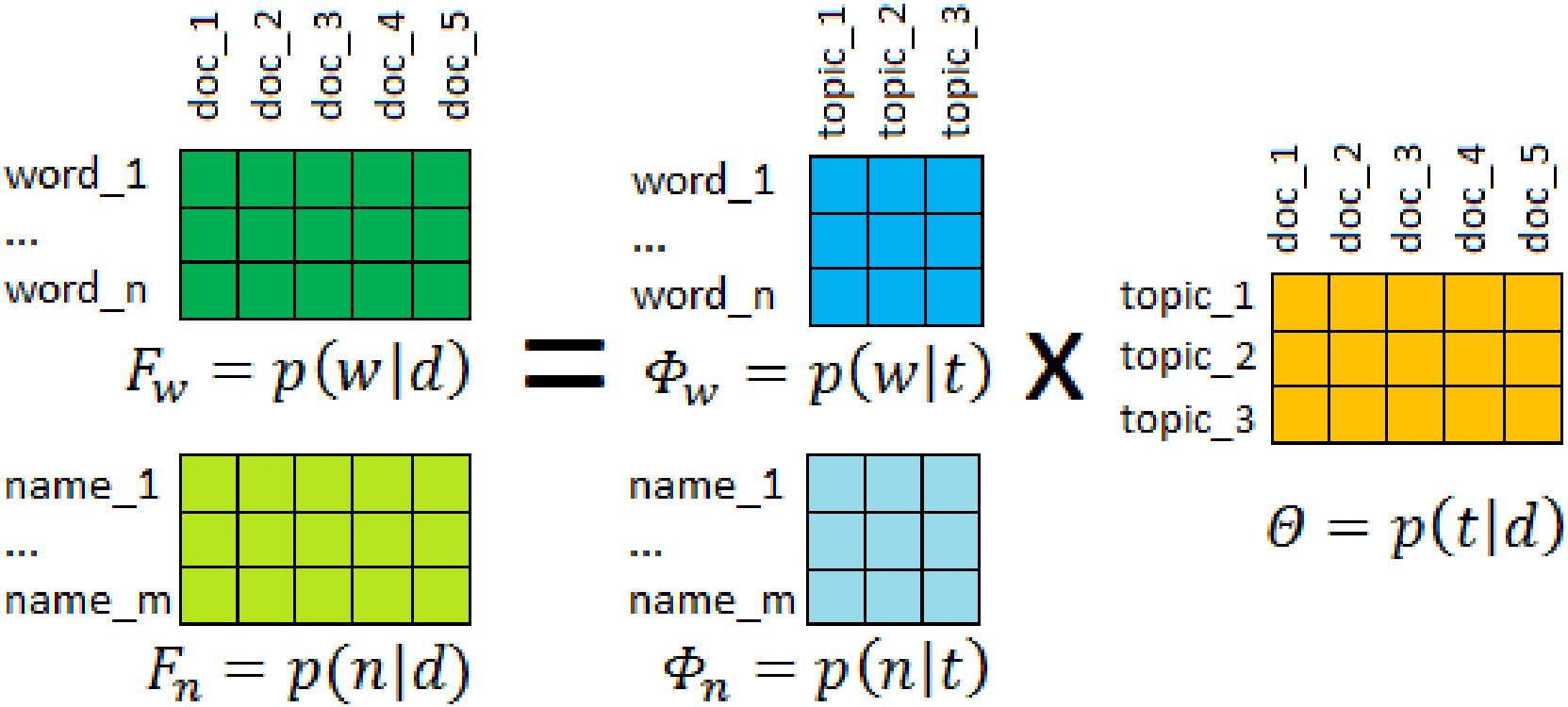
Выявление тематики документов $p(t|d)$ и терминов $p(w|t)$,
а также *модальностей*: $p(\text{автор}|t)$, $p(\text{время}|t)$, $p(\text{ссылка}|t)$,
$p(\text{тег}|t)$, $p(\text{баннер}|t)$, $p(\text{изображение}|t)$, $p(\text{пользователь}|t)$, ...



Мультимодальная тематическая модель

Пусть имеются две модальности:

- обычные слова
- имена авторов (категорий / тегов и т.п.)



Алгоритм обучения

Оффлайн EM-алгоритм

1. Многократное итерирование по коллекции.
2. Однократный проход по документу.
3. Необходимость хранить матрицу Θ.
4. Φ обновляется в конце каждого прохода по коллекции.
5. Применяется при обработке небольших коллекций.

Онлайн EM-алгоритм

1. Однократный проход по коллекции.
2. Многократное итерирование по документу.
3. Нет необходимости хранить матрицу Θ.
4. Φ обновляется через заданное число документов.
5. Применяется для больших коллекций в потоковом режиме.

Список регуляризаторов в BigARTM

BigARTM реализует мультимодальную ARTM.
Часто используемые регуляризаторы (можно добавлять свои)[1]:

1. SmoothSparseThetaRegularizer: сглаживание/разреживание Θ
2. SmoothSparsePhiRegularizer: сглаживание/разреживание Φ
3. DecorrelatorPhiRegularizer: декоррелирование тем в Φ
4. TopicSelectionThetaRegularizer: разреживания $p(t)$ и отбор тем
5. ImproveCoherencePhiRegularizer: повышение когерентности[2]

Полный список с описаниями — в онлайн-документации.

[1] названия классов в Python API

[2] мера качества, коррелирующая с экспертными оценками интерпретируемости

Список метрик качества в BigARTM

Часто используемые метрики качества[3] (можно добавлять свои):

1. PerplexityScore: перплексия
2. SparsityPhiScore: разреженность Φ
3. SparsityThetaScore: разреженность Θ
4. TopicKernelScore: характеристики ядер тем + когерентность[4]
5. TopTokensScore: наиболее вероятные в темах слова + когерентность

Полный список с описаниями — в онлайн-документации.

[3] названия классов в Python API

[4] *Vorontsov K. V., Potapenko A. A.* Tutorial on Probabilistic Topic Modeling: Additive Regularization for Stochastic Matrix Factorization. AIST 2014.

Важные особенности BigARTM

- Все регуляризаторы и метрики приспособлены для работы с мультимодальными моделями.
- Представление документа либо как «мешка слов», либо как последовательного текста $w_1, \ldots, w_{n_d}$.
- Считывание и модификация матрицы вспомогательных переменных $p_{tdi} = p(t|d, w_i)$ для любого документа.
- Построение иерархических тематических моделей.
- Чтение данных как с диска, так и из RAM.

Про входные форматы данных

BigARTM оперирует данными во внутреннем бинарном представлении, называемыми *батчами*.

Получить батчи из своих данных можно с помощью встроенного парсера, который поддерживает несколько типов входных форматов, основной — формат Vowpal Wabbit.

Батч — текстовый файл, каждая строка — один документ.

Формат строк:
[<title>] [|@default_class] {token_1[:counter_1]} {other modalities}

doc1 Alpha Bravo:10 Charlie:5 |@author Ola_Nordmann

doc2 Bravo:5 Delta Echo:3 |@author Ivan_Ivanov

Детальное описание форматов — в онлайн-документации.

Smooth/Sparse Φ

Формула M-шага:

$$\phi_{wt} = \underset{w \in W}{\text{norm}}\left(n_{wt} + \tau f(\phi_{wt}) d_w\right)$$

- Позволяет сглаживать/разреживать различные подмножества тем заданным распределением.
- Для контроля типа распределения по словам можно использовать *словарь* d_w и функцию f:
  1. Словарь d_w — это объект класса `Dictionary`, который содержит информацию о коллекции и дополнительные изменяемые множители для каждого слова.
  2. Функция f — некоторое преобразование, позволяющее текущему значению ϕ_{wt} влиять на собственную регуляризацию.

Что за функция f?

Напоминание:

$$\mathrm{KL}(P||Q) = \sum_{i=1}^{n} p_i \ln \frac{p_i}{q_i}$$

Регуляризатор сглаживания:

$$\sum_{t\in T} \mathrm{KL}(\beta_w||\phi_{wt}) \to \min_{\Phi} \quad \Leftrightarrow \quad R(\Phi) = \sum_{t\in T}\sum_{w\in W} \beta_w \ln \phi_{wt} \to \max_{\Phi}$$

$$\phi_{wt}\frac{\partial R}{\partial \phi_{wt}} = \beta_w \quad \Leftrightarrow \quad f(\phi_{wt}) = 1$$

Если $\ln(x)$ заменить на $\mu(x)$, то $f(x) = x\mu'(x)$.
В случае KL-дивергенции $\mu \equiv \ln$, поэтому $f(x) = 1$.

Стратегии использования регуляризатора

- Простое сглаживание/разреживание всех значений матрицы Φ заданным значением n: достаточно создать один регуляризатор и задать ему $\tau = n$.
- Разделение тем на две группы (предметные и фоновые), разреживать первую группу и сглаживать вторую.
  Для этого надо создать два регуляризатора и каждому заполнить соответствующее поле `topic_names`.
  У первого регуляризатора τ будет отрицательным, у второго — положительным.
- Сглаживание/разреживание только слов заданных модальностей: нужно создать один регуляризатор и заполнить его поле `class_ids`.

Стратегии использования регуляризатора

- Сглаживание/разреживание слов из заданного списка: для этого нужно внести правки в словарь, после чего указать этот словарь в качестве параметра `dictionary` регуляризатора.
- Разреживание/сглаживание с увеличением влияния маленьких значений ϕ_{wt} и уменьшением влияния больших. Для этого нужно создать объект функции `KlFunctioninfo` и передать его в качестве параметра `kl_function_info` регуляризатора (помните, что f — это производная от выбранной функции).

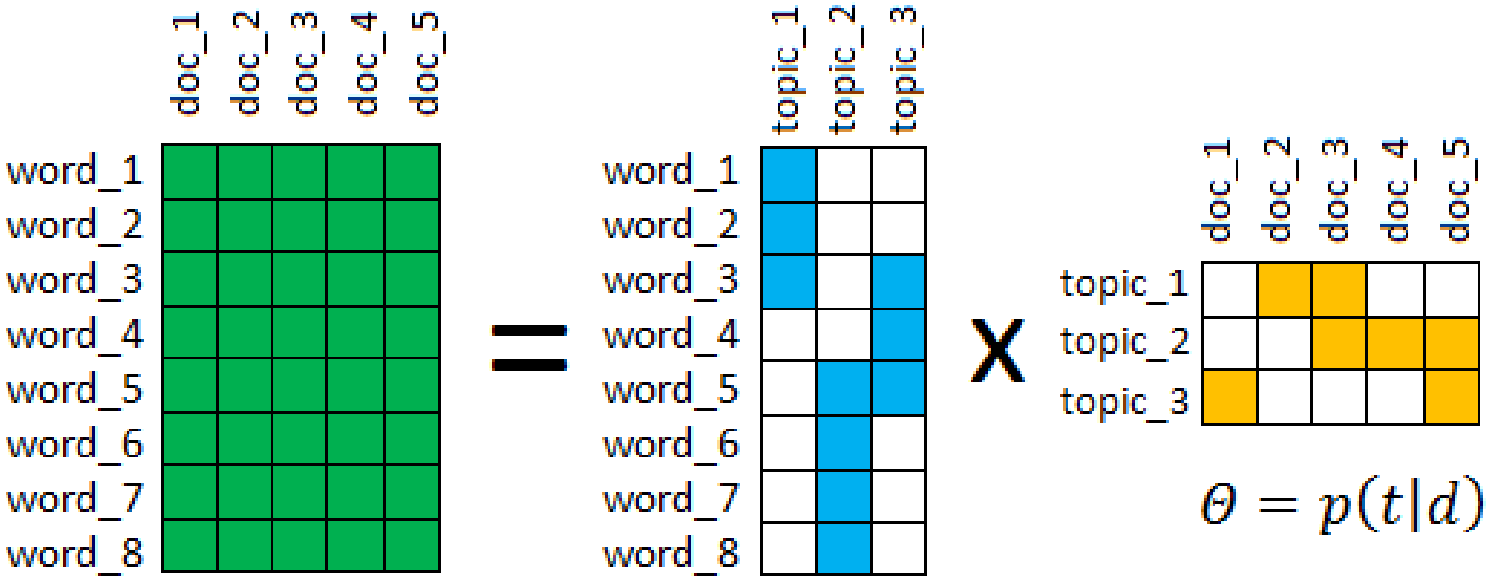
Стратегии использования регуляризатора

1. Все эти стратегии можно комбинировать и смешивать.
2. Сглаживание фоновых тем можно включать с первой итерации, при постоянном коэффициенте регуляризации.
3. Разреживание лучше начинать спустя некоторое число итераций, когда алгоритм уже почти сойдётся.
4. С помощью словарей и списков тем можно сглаживать/разреживать любые подматрицы Φ.

Пример использования разреживания

Естественные предположения:

- каждая тема состоят из небольшого числа слов
- темы, как множества слов, существенно различны
- каждый документ относится к небольшому числу тем

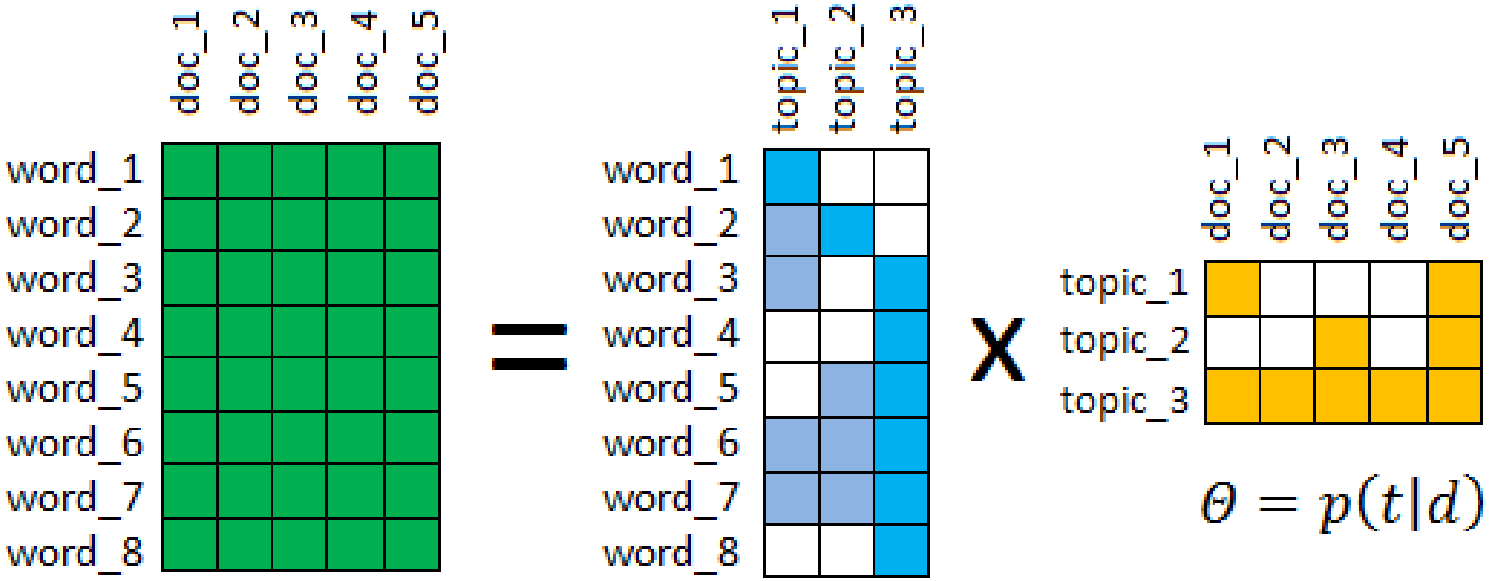


$F = p(w|d)$ $\Phi = p(w|t)$ $\Theta = p(t|d)$

Пример использования сглаживания

Частичное обучение тем по ключевым словам:

- для некоторых тем задаются *семантические ядра*
- для некоторых документов задаются темы
- для фоновых тем сглаживание по словарю общей лексики



Про словари в BigARTM

Словари в BigARTM играют огромную роль, они используются:

- для инициализации тематической модели
- для некоторых метрик качества
- для некоторых регуляризаторов

О словарях можно прочесть в нескольких разделах документации.

Словарь в Python можно сохранить на диск методом `artm.Dictionary.save_text(filename)`,

отредактировать и загрузить обратно двойственным методом `load_text()`.

Про словари в BigARTM

В текстовом виде Dictionary представляет собой набор строк, каждая строка (кроме первой заголовочной) соответствует одному уникальному слову из словаря коллекции.

Строка имеет следующий формат:

```
token   modality   value   tf   df
```

1. Первые два элемента — это само слово в виде строки и его модальность, последние два — значения tf и df данного слова. Все эти значения считаются библиотекой в процессе парсинга.

2. Поле `value` тоже считается при парсинге, и представляет собой нормированное значение tf. Но его можно переопределять. Оно используется в регуляризаторе SmoothSparsePhi как множитель коэффициента регуляризации для данного слова.

Smooth/Sparse Θ

Формула M-шага:

$$\theta_{td} = \underset{t \in T}{\text{norm}}\left(n_{td} + \tau \alpha_i f(\theta_{td}) m_{dt}\right)$$

- Позволяет сглаживать/разреживать различные подмножества тем заданным распределением
- Параметр α_i позволяет регулировать степень воздействия регуляризатора на данной внутренней итерации i
- Для контроля распределения по темам и по документам можно использовать:
  1. Вектор или матрицу m (о ней подробно написано в документации, работает как дополнительный множитель)
  2. Функция f позволяет текущему значению θ_{wt} влиять на свою регуляризацию

Стратегии использования регуляризатора

- Простое сглаживание/разреживание матрицы Θ.
- Разделение на предметные и фоновые темы.
- Использование функции f и параметра α (`alpha_iter`).
- Регуляризацию можно использовать при получении векторов θ_d для новых документов.

Не забывайте про флаги

- `cache_theta` — хранить Θ или нет
- `reuse_theta` — переиспользовать Θ с прошлой итерации или нет.

Decorrelator Φ

Формула M-шага:

$$\phi_{wt} = \underset{w \in W}{\text{norm}}\Big(n_{wt} - \tau \phi_{wt} \sum_{s \in T} \phi_{ws}\Big)$$

- Позволяет разреживать Φ таким образом, чтобы получать как можно более непохожие темы.
- Воздействие регулируется по темам и модальностям аналогично описанному ранее.
- Рекомендуется включать почти сразу после начала обучения.

О подборе параметров

Параметры бывают структурные:

- Число батчей и документов в батчах
- Число потоков-обработчиков
- Число проходов по коллекции/документу
- Тип алгоритма
- Параметры алгоритма (если онлайн)

Или обычные:

- Наборы регуляризаторов и их параметров
- Наборы модальностей и их параметров

Подбор структурных параметров

- Число потоков обработчиков выбирается исходя из возможностей экспериментальной машины
- Число батчей должно быть кратно числу потоков
- Размер батча — не слишком маленьким, но и не слишком большим (порядка 10^5 слов)
- Тип алгоритма — оффлайн проще, онлайн — круче.
- Параметры алгоритма — чёткой методики нет, можно перебором.
- Число тем — регуляризатор отбора тем или априорные предпочтения.

Подбор траектории регуляризации

Не надо добавлять в модель сразу все регуляризаторы!

Легче добавлять по одному, оптимизируя τ.
При этом надо всегда понимать, зачем именно регуляризатор добавляется в модель и как он примерно работает.

- Сглаживание/разреживание.
- Декоррелятор.
- Частичное обучение.
- Модальности.

Подбор параметров: grid search или random search.

Относительные коэффициенты регуляризации (**Медленнее!**):
gamma=0.5 — можно перебирать τ от 0 до 1 (только Φ).

Что нужно для эксперимента, кроме BigARTM

Помимо BigARTM, установленного и настроенного под Python, желательно пользоваться следующими инструментами:

- Jupyter Notebook
- Лемматизаторы (pymorphy2, pymystem)
- Базовые средства обработки текстов из nltk
- Модули numpy, pandas, re и matplotlib
- Программы для просмотра больших текстовых файлов (Windows: emeditor, Linux/MacOS: less)

Какие бывают типы результатов

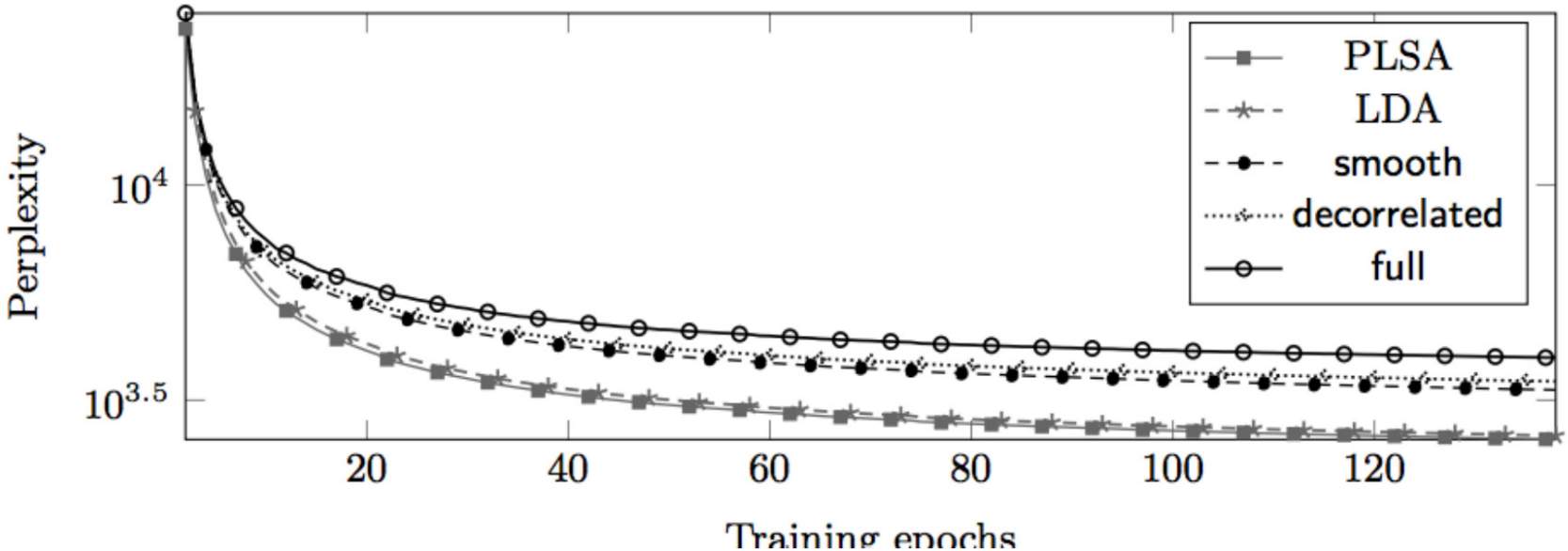
- Перплексия и другие числовые метрики.
- Топ-слова в темах.
- Документы (топ-документы надо извлекать).

Извлечение топ-документов для большой коллекции:

1. Обучили модель без сохранения Θ.
2. Идём в цикле по батчам и подаём их в `ARTM.transform()` (просим извлечь `dense_theta`).
3. Получив Θ для очередного батча, анализируем её (максимум по столбцам, например).
4. Закончив обработку, удаляем Θ для текущего батча, переходим к следующему.

Графики

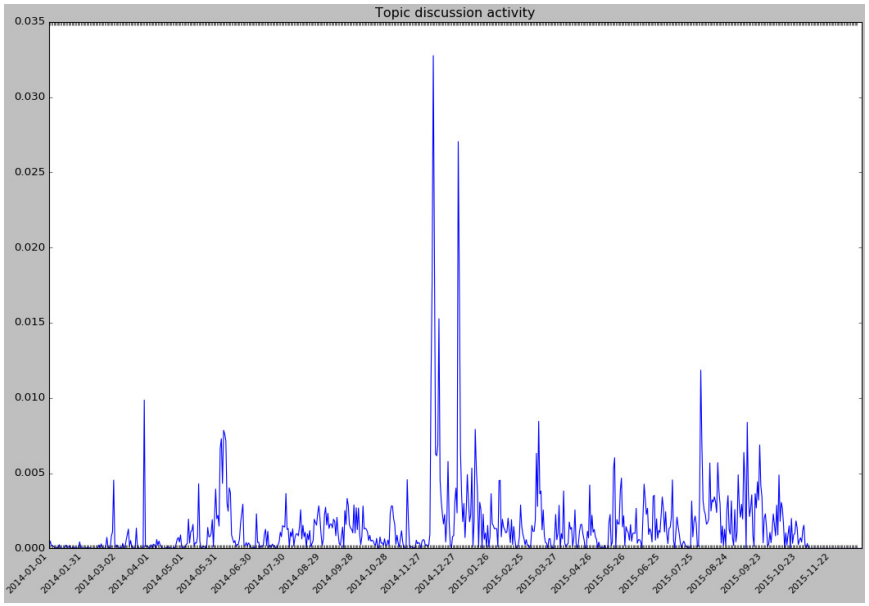
важны для понимания и презентации числовых метрик:



Топ-слова и документы придётся просматривать глазами.

Презентация результатов

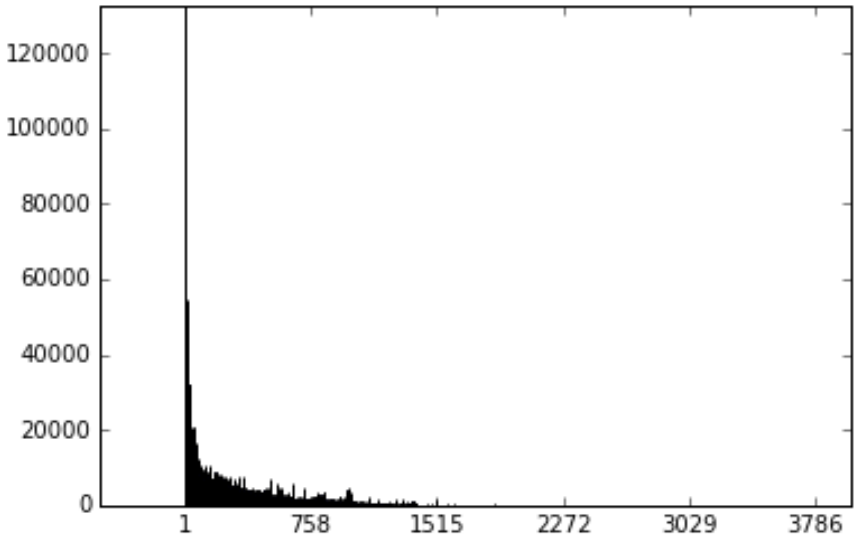
Для визуализации специальных модальностей можно пользоваться разнообразными инструментами



- Метки времени — график распределения $P(\text{время}|t)$
- Геотеги — наложить на реальную карту

Презентация результатов

Гистограммы полезны для оценивания частот модальностей.



- Важно: визуализация всегда нагляднее других способов (хоть и не всегда полнее).
- В Python много средств визуализации.
  Почти всегда можно подобрать что-то с нуля за 2-3 часа.
- Можно рисовать в Matlab или в LaTeX (tikz)

Постановка задачи

Дано:

- Коллекция постов сайта LiveJournal.
- Словарь этнонимов (слов, связанных с этносами).

Задача: выявить как можно большее количество качественных тем, связанных с этно-проблемами.

Метрика качества: оценки асессоров.

Параметры коллекции

Параметры коллекции:

- 1.58 млн. документов в виде «мешка слов»;
- 860 тыс. слов словаре;
- коллекция прошла лемматизацию.

Особенности:

- много слов с ошибками;
- коллекция русскоязычная, но присутствую термины на английском, украинском;
- много жаргонных слов и терминов специфических областей — сложно понимать и интерпретировать темы!

Подготовка данных

Парсим данные в формат Vowpal Wabbit.

Сохраним только те слова, которые:

1. содержат только символы киррилицы и дефис;
2. содержат не более одного дефиса (встречаются слова вроде --, ----);
3. имеют длину не менее 3 символов (встречаются слова вроде 'а', 'ж');
4. встречаются в коллекции не менее 20 раз;

Объём итогового словаря: 90 тыс слов.

В таких случаях бывают полезны регулярные выражения.

Составление словаря этнонимов

Описание проблемы:

- Имеется словарь из нескольких сотен этнонимов
- Слова собраны в списки (например [абхаз, абхазец, абхазка])
- Часть этих слов не встречаются в LJ
- Нужно составить аналогичный словарь, специфичный для LJ

Можно сделать вручную:

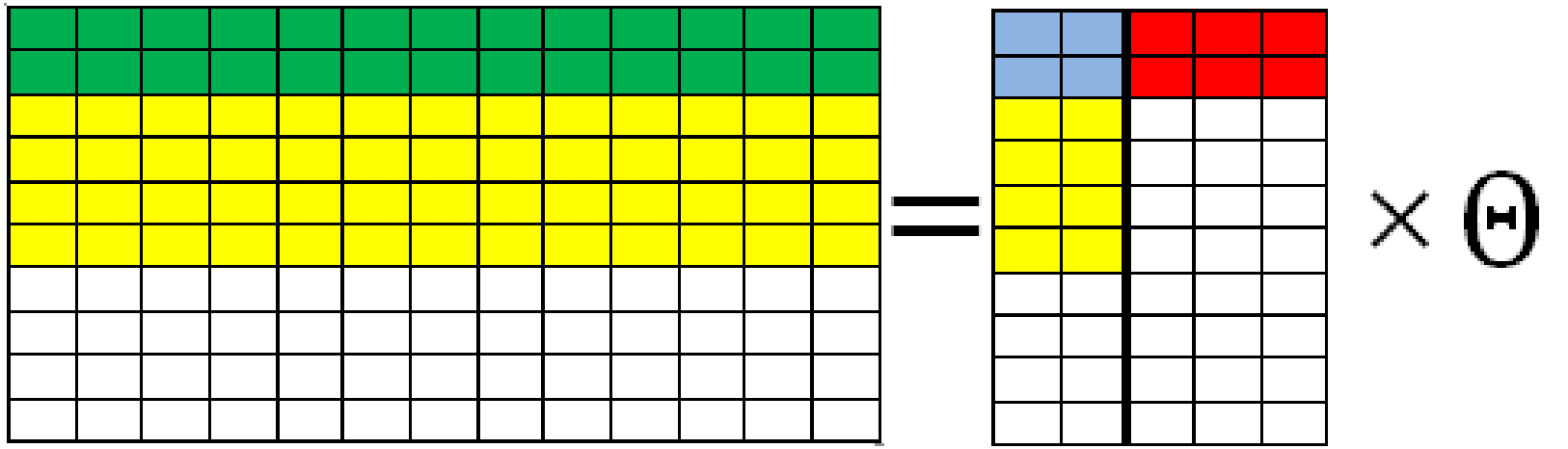
1. преобразовать списки всех слов в один линейный список;
2. пройтись по этому списку и для каждого слова найти все максимально похожие на него;
3. выбрать вручную в получившемся множестве все наиболее этничные слова, по 1-2 на каждый этноним исходного списка.

Объём итогового словаря этнонимов: 250 слов.

Примеры этнонимов

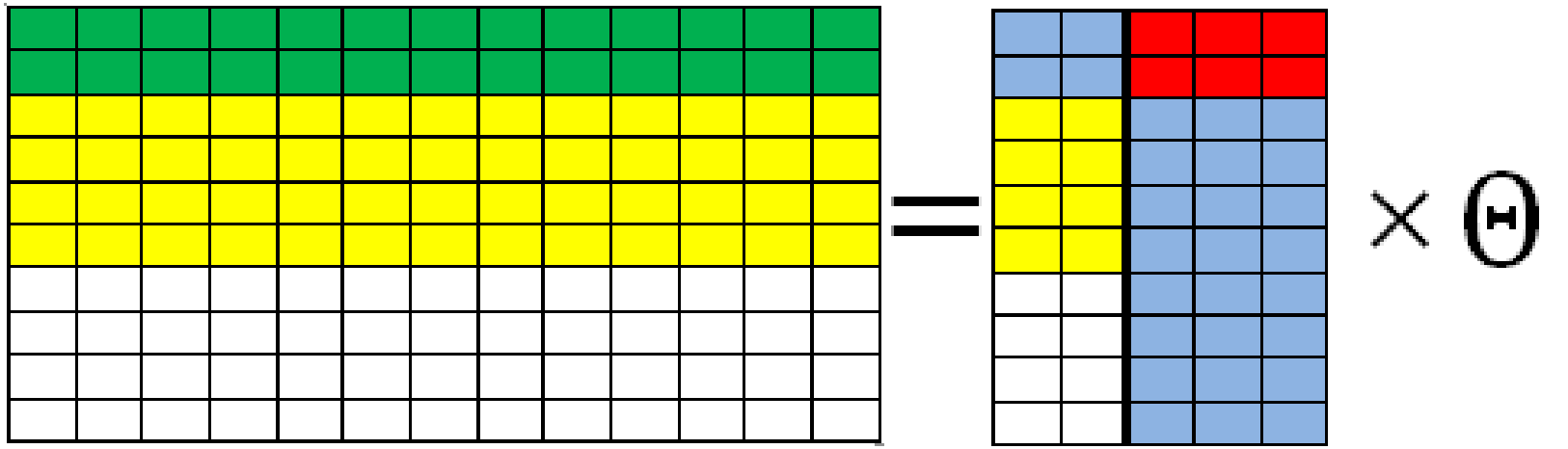
османский	русич
восточноевропейский	сингапурец
эвенк	перуанский
швейцарская	словенский
аланский	вепсский
саамский	ниггер
латыш	адыги
литовец	сомалиец
цыганка	абхаз
ханты-мансийский	темнокожий
карачаевский	нигериец
кубинка	лягушатник
гагаузский	камбоджиец

Сглаживание/разреживание этнонимов



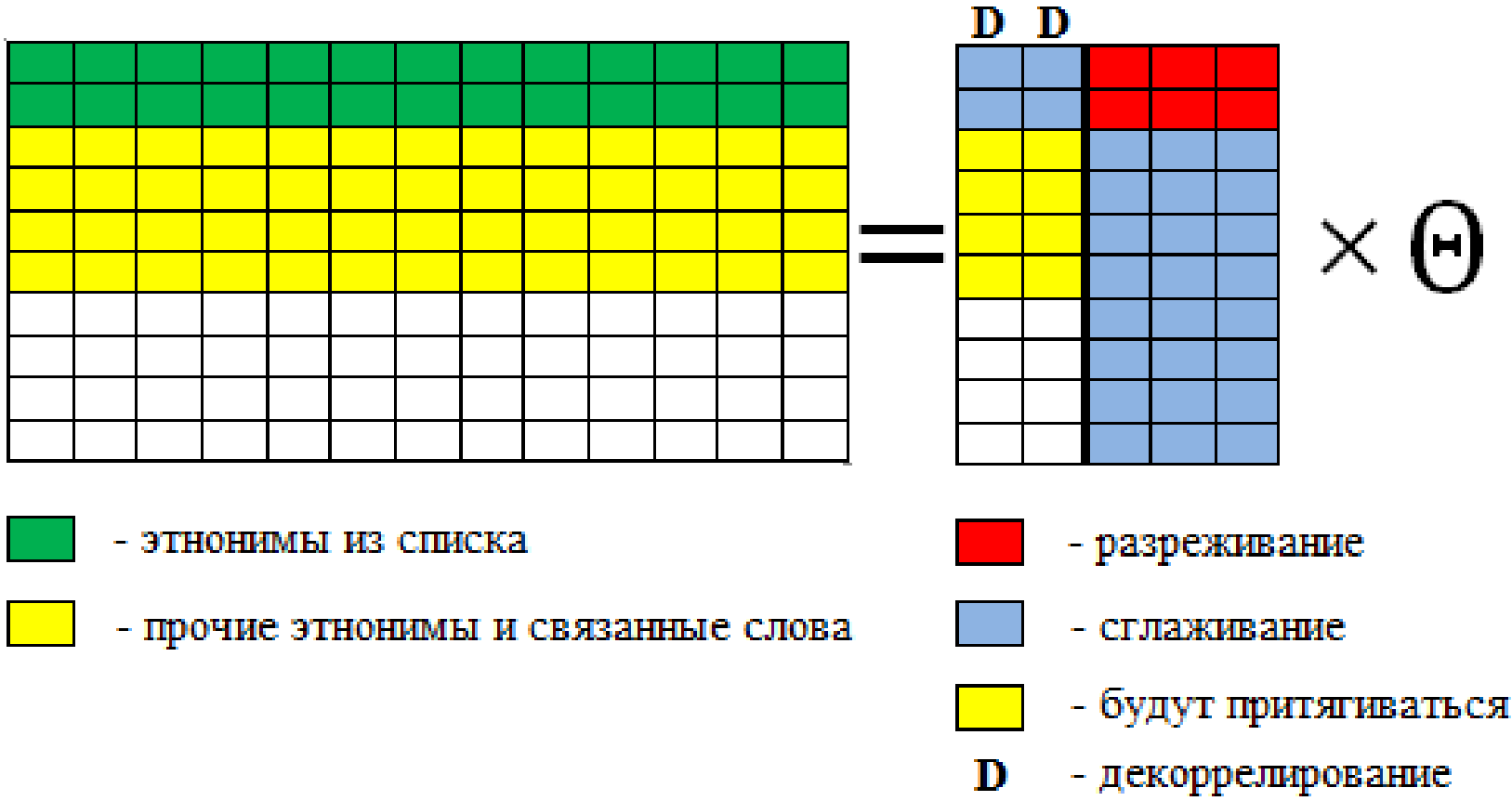
- этнонимы из списка
- прочие этнонимы и связанные слова
- разреживание
- сглаживание
- будут притягиваться

+ сглаживание обычных слов

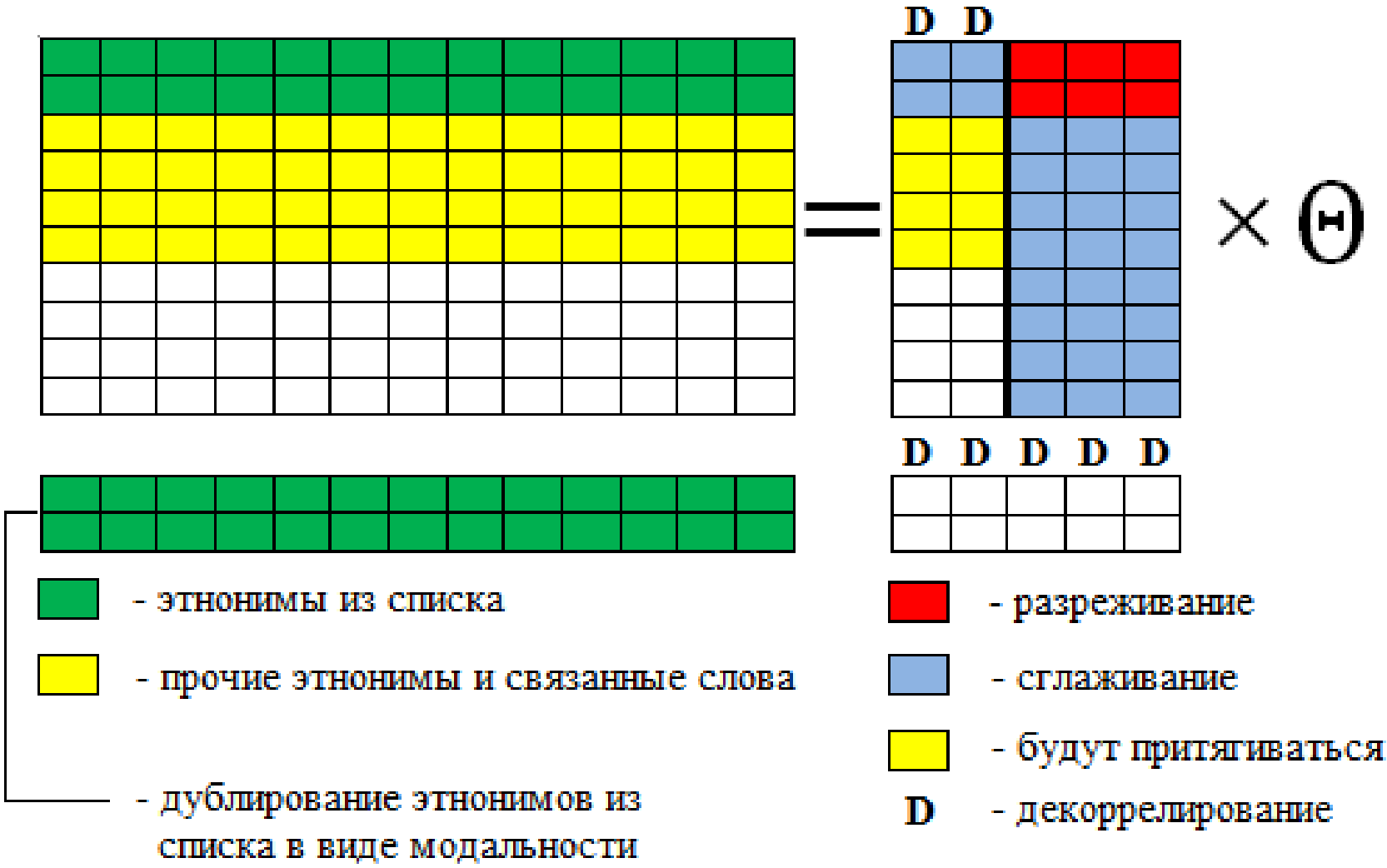


- этнонимы из списка
- прочие этнонимы и связанные слова
- разреживание
- сглаживание
- будут притягиваться

+ декорреляция этничных тем



+ модальность этнонимов



Примеры лучших тем

(русские): акция, организация, митинг, движение, активный, мероприятие, совет, русский, участник, москва, оппозиция, россия, пикет, протест, проведение, националист, поддержка, общественный, проводить, участие,
(славяне, византийцы): славянский, святослав, жрец, древние, письменность, рюрик, летопись, византия, мефодий, хазарский, русский, азбука,
(сирийцы): сирийский, асад, боевик, район, террорист, уничтожать, группировка, дамаск, оружие, алесио, оппозиция, операция, селение, сша, нусра, турция,
(турки): турция, турецкий, курдский, эрдоган, стамбул, страна, кавказ, горин, полиция, премьер-министр, регион, курдистан, ататюрк, партия,
(иранцы): иран, иранский, сша, россия, ядерный, президент, тегеран, сирия, оон, израиль, переговоры, обама, санкция, исламский,
(палестинцы): террорист, израиль, терять, палестинский, палестинец, террористический, палестина, взрыв, территория, страна, государство, безопасность, арабский, организация, иерусалим, военный, полиция, газ,
(ливанцы): ливанский, боевик, район, ливан, армия, террорист, али, военный, хизбалла, раненый, уничтожать, сирия, подразделение, квартал, армейский,
(ливийцы): ливан, демократия, страна, ливийский, каддафи, государство, алжир, война, правительство, сша, арабский, али, муаммар, сирия,
(евреи): израиль, израильский, страна, израил, война, нетаньяху, тель-авив, время, сша, сирия, египет, случай, самолет, еврейский, военный, ближний,

Некоторые результаты

Модель	Лучших тем	Хороших тем	Удовл. тем	Всего
PLSA (300)	9	11	18	38
PLSA (400)	12	15	17	44
С.Р.Д. (200+100)	18	33	20	71
С.Р.Д. (250+150)	21	27	20	68
С.Р.Д.М. (300+100)	28	23	23	74
С.Р.Д.М. (250+150)	22	25	33	80
С.Р.Д.М. (250+150) (после настройки)	32	42	40	104

С – сглаживание, Р – разреживание,
Д - декорреляция, М – этномодальность

Что можно делать ещё?

- Эти эксперименты были продолжены на более крупной и сложной коллекции IQBuzz постов разных русскоязычных социальных медиа (в основном Вконтакте).
- Был вручную собран новый, более полный и насыщенный существительными словарь этнонимов.
- Постановка задачи была усложнена: в дополнение для каждой релевантной темы требовалось исследовать её изменение в пространстве и времени.
- Для этого строились мультимодальные модели с дополнительными модальностями геотегов авторов, а также меток времени публикации сообщения.

Пример темы с привязкой ко времени и пространству

Топ-слова:
чеченский, чечня, кадыров, боевик, террорист, убийство, рамзан, грозный, спецназ, наемник, кавказ, погибать, операция, теракт, вооруженный, боевой, заложник, дудаев, лидер, командир

Топ-геотеги:
Москва, Санкт-Петербург, Чечня

Топ-метки времени:
Сосредоточены в начале и конце декабря 2014

Комментарий:
Совпадает с датой 20-тилетия начала войны в Чечне.

Данные IQBuzz охватывают период 2014-2015 годов.

Тем такого же качества — больше 10% от общего количества и примерно 30% от общего числа признанных этничными.

Спасибо за внимание! :)

bigartm.org