



NATIONAL RESEARCH  
UNIVERSITY

Laboratory for Social and Cognitive  
Informatics,  
St. Petersburg School of Physics,  
Mathematics, and Computer  
Science, Department of Informatics



## TOPIC MODELING

Koltcov S.

Saint Petersburg, 2024



## The contents of the lectures

- Введение в проблематику тематического моделирования.
- Обзор математических принципов тематического моделирования:
  - А. Метод максимизации функции правдоподобия с помощью алгоритма Е-М: Модель 'Probabilistic Latent Semantic Analysis, PLSA'.  
Модель 'Latent Dirichlet allocation'.
- В. метод оценки математического ожидания с помощью выборки Гиббса.
- Краткое описание возможностей и ограничений тематического моделирования Проблема выбора количества тем. Проблема устойчивости тематического моделирования.
- Математические основы проблемы ТМ Меры качества тематической модели. Примеры тематических моделей в социальных науках.



## Introduction to the problems of topic modeling

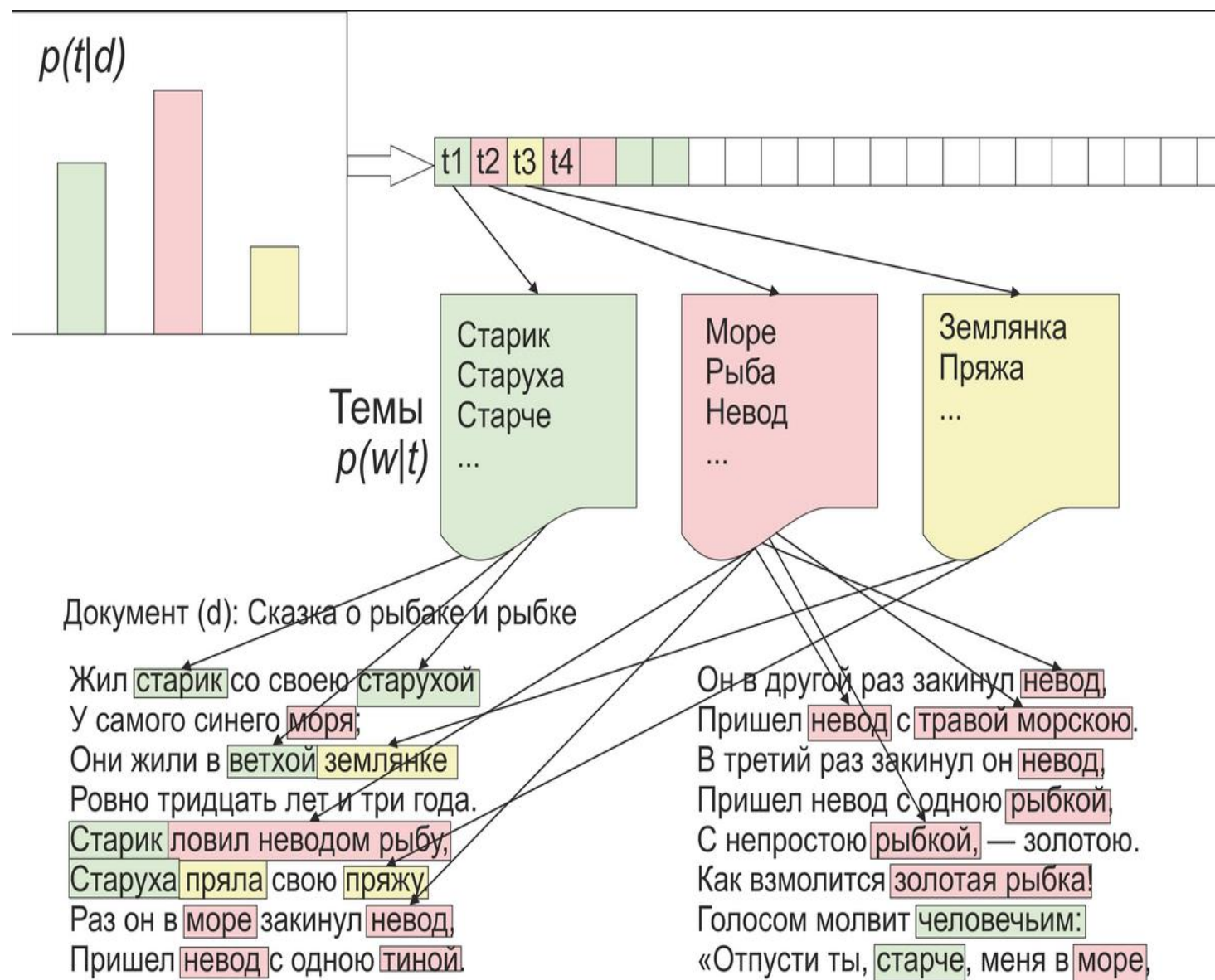
**Marketing:** тематический анализ отзывов о товарах и услугах. Выявление аспектов для анализа настроения отзывов (аспектный анализ настроения на основе LDA).

**Sociology:** "Народная" повестка дня и общественное мнение онлайн на основе пользовательского контента в социальных сетях. Корреляция повестки дня с демографическими характеристиками, событиями и внешними повестками дня (например, в СМИ)

**Media research:** тематическая структура медиаконтента. Разница между изданиями, цензура, динамика роста и падения внимания к новостным сюжетам.

**Scientometry:** изучение структуры научного знания и его эволюции. Составление обзоров литературы.

## Probabilistic problem statement for topic modeling



Topic model — модель коллекции текстовых документов, определяющая, к каким темам относится каждый документ в коллекции. Алгоритм построения тематической модели: на вход модели поступает коллекция текстовых документов. На выходе для каждого документа получается числовой вектор, состоящий из оценок степени принадлежности этого документа к каждой из тем.



## Probabilistic problem statement for topic modeling

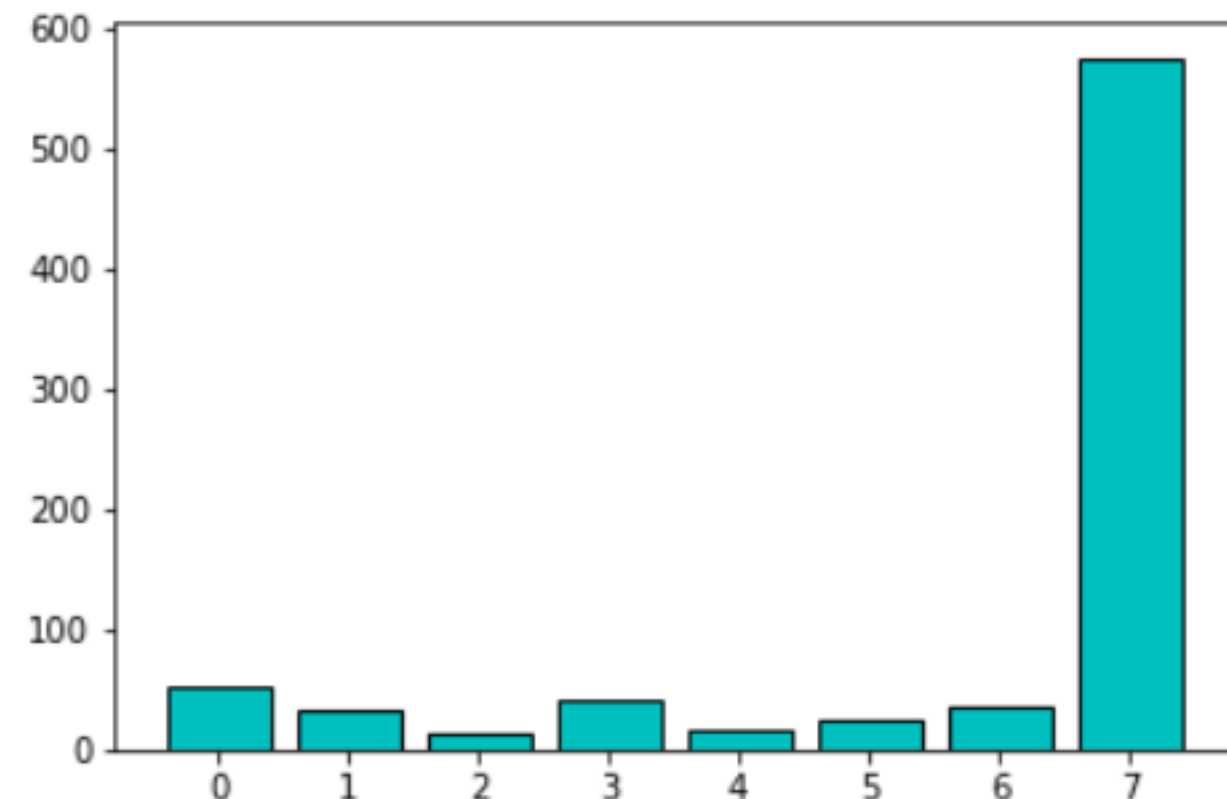
Пусть  $D$  - набор текстовых документов, а  $W$  - множество (словарь) всех уникальных слов. Каждый документ  $d \in D$  - это последовательность терминов  $w_1, \dots, w_d$  из словаря ( $W$ ). Мы также предполагаем, что существует конечное множество тем  $T$ , и каждое вхождение слова  $w$  в документ  $d$  связано с некоторой темой  $t \in T$ . Коллекция документов рассматривается как случайная и независимая выборка троек  $(w_i, d_i, t_i)$ ,  $i = 1, \dots, n$  из дискретного распределения  $p(w, d, t)$  на конечном вероятностном пространстве  $W \times D \times T$ . Слова  $w$  и документы  $d$  - наблюдаемые переменные, субъект  $t \in T$  - латентная (скрытая) переменная. **Предполагается, что порядок терминов в документах не важен для идентификации субъекта (мешок слов). Порядок документов в коллекции также не имеет значения.**



## Matrix $\Theta$ - topics distribution in documents

[8]:

	Topic0	Topic1	Topic2	Topic3	Topic4	Topic5	Topic6	dominant_topic
Doc0	0.010000	0.010000	0.920000	0.010000	0.010000	0.010000	0.010000	2
Doc1	0.010000	0.010000	0.010000	0.010000	0.670000	0.010000	0.280000	4
Doc2	0.010000	0.010000	0.790000	0.010000	0.010000	0.010000	0.140000	2
Doc3	0.020000	0.020000	0.020000	0.020000	0.020000	0.020000	0.900000	6
Doc4	0.940000	0.010000	0.010000	0.010000	0.010000	0.010000	0.010000	0
Doc5	0.010000	0.010000	0.010000	0.010000	0.010000	0.010000	0.910000	6
Doc6	0.010000	0.010000	0.010000	0.010000	0.640000			
Doc7	0.010000	0.010000	0.010000	0.010000	0.010000			
Doc8	0.020000	0.020000	0.020000	0.020000	0.020000			
Doc9	0.010000	0.630000	0.010000	0.010000	0.010000			
Doc10	0.010000	0.010000	0.010000	0.010000	0.010000			
Doc11	0.020000	0.020000	0.020000	0.020000	0.020000			
Doc12	0.010000	0.010000	0.770000	0.010000	0.010000			
Doc13	0.020000	0.020000	0.020000	0.020000	0.910000			
Doc14	0.010000	0.010000	0.010000	0.930000	0.010000			





## Matrix $\Phi$ – word distribution in topics

[75]:

	Topic 0	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
Word 0	что - 0.0004698091106239771	турции - 0.0003081567456290563	что - 0.0005595929984206749	кино - 0.0008377494184893066	миронова - 0.00038785660283128723	тамара - 0.0002899678330191077
Word 1	было - 0.00045069188824187873	новости - 0.00026218325858764395	для - 0.0004909673222097849	актёров - 0.0006882717680612737	что - 0.0003523639008715388	эвра - 0.0002814785833549309
Word 2	сама - 0.0003799606292472247	риа - 0.000260257133978759	только - 0.000415914812551397	кто - 0.0006281071291426114	платт - 0.0003287332341951211	меньшов - 0.00027926902209925825
Word 3	гостями - 0.0003733567480631264	евстигнеев - 0.0002398973477305586	больше - 0.00038533637021255035	программа - 0.0006129727273055995	михайлов - 0.00032093511409124026	прохоренко - 0.00027534367118077073
Word 4	становятся - 0.00037335455688781603	россиян - 0.00023919681240274778	крымского - 0.00036362016932275297	жизни - 0.0006116706036697333	теличкина - 0.0003084961624621251	её - 0.000271491447661209
Word 5	люди - 0.00036664935894809535	nacional - 0.0002301310751217433	волчек - 0.0003462529584878737	что - 0.0005900216353031027	георгий - 0.00026916693703453746	пирог - 0.0002561178331886546
Word 6	обществе - 0.0003566117509430164	el - 0.00023012959928839767	момента - 0.0003418114960347403	тех - 0.000554808273682613	павел - 0.00026774550827918497	зелёная - 0.0002560817262403854
Word 7	чьа - 0.0003565912094529661	игил - 0.00021421495451066618	уно - 0.0003418045940863241	нашего - 0.000541571646685365	никогда - 0.00026303024040720085	рина - 0.00025608118940324365
Word 8	личность - 0.00035658720458574473	ван - 0.0002018472458170297	дейр - 0.00033838791544775	ссср - 0.0005348414317714289	большая - 0.00025981700043984133	рины - 0.0002560801109106136
Word 9	вызывают - 0.0003565816472557962	болдуина - 0.00019822969960403715	борисов - 0.00029495745286122865	авторской - 0.0004785189503699558	мало - 0.00025897457522792506	сёмина - 0.00022792832325523838
Word 10	его - 0.00033809876867681556	алеппо - 0.00019102189597331812	украины - 0.00029426321946377225	людях - 0.0004717950638932046	был - 0.0002484538784704299	первого - 0.00022549558660126703
Word 11	ведущих - 0.0003329551412600506	жан - 0.00019073057360025886	мост - 0.00029404605645420325	всегда - 0.00046309333331577866	как - 0.0002483347595859574	алексиевич - 0.00022017162369273883
Word 12	интерес - 0.00032457711727933766	новогоднюю - 0.00018736058971367334	вертолет - 0.00029095614132396076	программы - 0.0004629812043686788	героя - 0.00024431115075935903	нефти - 0.0002183796027773021



## Mathematical model for topic modeling

Вероятность  $p(w|d)$  появления терминов  $w$  в документах  $d$  может быть выражена как произведение распределений  $p(w|t)$  и  $p(t|d)$ . Согласно формуле полной вероятности и гипотезе условной независимости, мы имеем следующее выражение:

$$p(w|d) = \sum_{t \in T} p(w|t)p(t|d) = \sum_{t \in T} \phi_{wt} \theta_{td}$$

где  $p(w|t)$  - распределение слов по темам, а  $p(t|d)$  - распределение документов по темам. Построение тематической модели данных означает решение обратной задачи, в которой необходимо найти множество скрытых тем  $T$ , то есть множество одномерных условных распределений  $p(w|t) \equiv \phi(w, t)$  для каждой темы  $t$ , составляющих матрицу  $\Phi$  (распределение слов по темам), и множество одномерных распределений  $p(t|d) \equiv \theta(t, d)$  (матрица  $\theta$ , распределение документов по темам) для каждого документа  $d$  на основе наблюдаемых переменных  $d$  и  $W$ .





## The main directions of topic modeling

1. Модели, основанные на методе максимизации логарифмического правдоподобия. Математическая формулировка этого семейства моделей заключается в том, что матрица распределения слов для документов  $F$  представляется как произведение матрицы распределения слов для тем  $\Phi$  и матрицы распределения тем для документов  $\Theta$ . Таким образом, задача  $F = \Phi\Theta$  сводится к поиску стохастического разложения матрицы, которое выполняется с помощью алгоритма Е-М. Для нахождения приближенного решения максимизируется логарифм правдоподобия.

$$p(w|d) = \sum_{t \in T} p(w|t)p(t|d) = \sum_{t \in T} \phi_{wt} \theta_{td}$$

2. Модели, основанные на методе Монте-Карло. При таком подходе вероятность распределения слов и документов можно оценить с помощью математического ожидания, если знать подынтегральные функции. В качестве подинтегральных функций используются мультиномиальные функции и функции Дирихле. Скрытые распределения определяются с помощью процедуры выборки Гиббса.

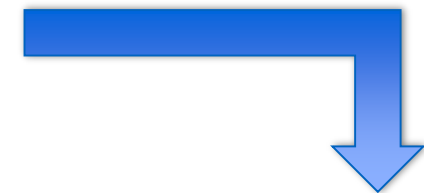
$$p(w | z, \beta) = \int p(w | z, \Phi) p(\Phi | \beta) d\Phi \quad p(z | \alpha) = \int p(z | \Theta) p(\Theta | \alpha) d\Theta$$



## Recovery of matrices $\Phi$ and $\Theta$ by maximizing the likelihood function using the E-M algorithm (PLSA).

Задача максимального правдоподобия формулируется следующим образом. Заменяя  $p(w|t)$  и  $p(t|d)$  матрицами  $\phi_{wt}$ ,  $\theta_{td}$  и прологарифмировав выражение, чтобы избавиться от произведения, мы получим выражение, в котором нужно подобрать значения матриц так, чтобы это выражение было максимальным.

$$p(w|d) = \sum_{t \in T} p(w|t)p(t|d) = \sum_{t \in T} \phi_{wt} \theta_{td}$$



$$\sum_{w \in W} \phi_{wt} = 1$$

$$L(\phi, \theta) = \sum_{d \in D} \sum_{w \in d} n_{wd} \ln \sum_{t \in T} \phi_{wt} \theta_{td} \rightarrow \max$$

Поиск максимального значения функции  $L(\phi, \theta)$  осуществляется с помощью EM-алгоритма.

## Algorithm for finding unknown distributions (E-M algorithm)

Для решения задачи PLSA использует итерационный процесс, в котором каждая итерация состоит из двух шагов - Е (ожидание) и М (максимизация) Перед первой итерацией выбирается начальное приближение параметров  $\phi_{wt}$ ,  $\theta_{td}$ .

**На шаге Е** условные вероятности  $p(t|d, w)$  всех тем  $t \in T$  для каждого термина  $w \in d$  в каждом документе  $d$  вычисляются по формуле Байеса с использованием текущих значений параметров  $\phi_{wt}$ ,  $\theta_{td}$ :

$$p(t | d, w) = \frac{p(w | t)p(t | d)}{p(w | d)} = \frac{\phi_{wt}\theta_{td}}{\sum_{s \in T} \phi_{ws}\theta_{sd}}. \quad H_{dwt} \equiv p(t | d, w) = \frac{\phi_{wt}\theta_{td}}{\sum_s \phi_{ws}\theta_{sd}}.$$

**На М-шаге**, наоборот, новое приближение параметров  $\phi_{wt}$ ,  $\theta_{td}$  вычисляется из условных вероятностей тем  $p(t | d, w)$ . Чтобы получить приближенное решение М-шага, запишем лагранжиан задачи

$$\mathcal{L}(\Theta, \Phi) = \sum_{d \in D} \sum_{w \in d} n_{dw} \ln \sum_{t \in T} \phi_{wt}\theta_{td} - \sum_{t \in T} \lambda_t \left( \sum_{w \in W} \phi_{wt} - 1 \right) - \sum_{d \in D} \mu_d \left( \sum_{t \in T} \theta_{td} - 1 \right).$$



## Algorithm for finding unknown distributions (E-M algorithm)

Продифференцировав  
лагранжиан по  $\phi_{wt}$  и приравняв  
нулю производную, получим:

$$\phi_{wt} = \frac{\sum_{d \in D} n_{dw} H_{dwt}}{\sum_{w' \in W} \sum_{d \in D} n_{dw'} H_{dw't}}.$$

где:

$$\phi_{wt} = \frac{\hat{n}_{wt}}{\hat{n}_t}, \quad \hat{n}_t = \sum_{w \in W} \hat{n}_{wt}, \quad \hat{n}_{wt} = \sum_{d \in D} n_{dw} H_{dwt}.$$

Проделав аналогичные действия с производной лагранжиана по  $\theta_d$ , получим:

$$\theta_{td} = \frac{\hat{n}_{dt}}{\hat{n}_d}, \quad \hat{n}_d = \sum_{t \in T} \hat{n}_{dt}, \quad \hat{n}_{dt} = \sum_{w \in W} n_{dw} H_{dwt}.$$

Если рассматривать коллекцию документов как выборку *токенов* — троек  $(d, w, t) \in D \times W \times T$ , то  $n_{dwt} = n_{dw} H_{dwt}$  есть оценка числа вхождений термина  $w$  в документе  $d$ , связанных с темой  $t$ . Соответственно, переменные  $\hat{n}_*$  интерпретируются как *счетчики* числа токенов:  $\hat{n}_{dt}$  — оценка числа токенов в документе  $d$ , связанных с темой  $t$ ;  $\hat{n}_{wt}$  — оценка числа токенов термина  $w$ , связанных с темой  $t$ ;  $\hat{n}_t$  — оценка общего числа токенов в коллекции, связанных с темой  $t$ ;  $\hat{n}_d = n_d$ ,



## Algorithm for finding unknown distributions (E-M algorithm)

Схема моделирования E – M алгоритма.

1. Инициализируем матрицы  $\phi_{wt}$ ,  $\theta_{td}$  (случайными числами в диапазоне [0 -1])
2. E-step. Вычисляем скрытые распределения, то есть вероятности слова в теме, и в документе.

$$p(t | d, w) = \frac{p(w | t)p(t | d)}{p(w | d)} = \frac{\varphi_{wt}\theta_{td}}{\sum_{s \in T} \varphi_{ws}\theta_{sd}}.$$

3. M-step. На данном этапе из известного распределения  $p(t|d, w)$  генерируем тему для текущего слова, после чего обновляем счетчики:

$$\hat{n}_{wt} \quad \hat{n}_{dt}$$

4. После этого пересчитываем матрицы  $\phi_{wt}$ ,  $\theta_{td}$

$$\phi_{wt} = \frac{\hat{n}_{wt}}{\hat{n}_t}, \quad \theta_{td} = \frac{\hat{n}_{dt}}{\hat{n}_d},$$

5. Переходим в пункт 2

Таким образом, прогоняем много раз все документы и все слова по этому циклу и получаем итоговые матрицы  $\phi_{wt}$ ,  $\theta_{td}$



## Latent Dirichlet Allocation (LDA)

Идея Блея заключается в следующем. Он предположил, что столбцы матрицы  $\phi_{wt}$  и строки  $\theta_{td}$  в модели PLSA генерируются распределениями Дирихле. Каждое распределение характеризуется своим параметром  $\alpha$  (для слов) и  $\beta$  (для документов), т. е. все параметры  $\alpha$  и  $\beta$  различны. В модели также предполагается, что вероятность встречи с каждой темой соответствует мультиномиальному распределению:

Multinomial distribution	$P(x   p) = \frac{n!}{\prod_{i=1}^K x_i!} \prod_{i=1}^K p_i^{x_i}$	Dirichlet distribution	$P(p; \alpha) = \frac{1}{B(\alpha)} \prod_{i=1}^K p_i^{\alpha_i - 1}$
-----------------------------	--	---------------------------	---

Эти предположения значительно упрощают байесовский вывод модели благодаря свойствам сопряженности распределений Дирихле с мультиномиальным распределением. Модель PLSA с функциями Дирихле преобразуется в следующую модель LDA:

$$\theta_{td} = \frac{n_{td} + \alpha_t}{n_d + \alpha_0}$$

$$\phi_{tw} = \frac{n_{wt} + \beta_w}{n_t + \beta_0}$$



## LDA на основе Gibbs sampling

Термины:  $D$  - пространство документов,  $W$  - пространство слов,  $Z$  - пространство тем. Темы - это скрытые параметры, которые необходимо найти. И оценка основывается на двух вещах: 1. Оценка проводится в соответствии с ожиданиями. 2. Используются мультиномиальные функции и функции Дирихле.

$$p(w | z, \beta) = \int p(w | z, \Phi) p(\Phi | \beta) d\Phi$$

$$p(z | \alpha) = \int p(z | \Theta) p(\Theta | \alpha) d\Theta$$

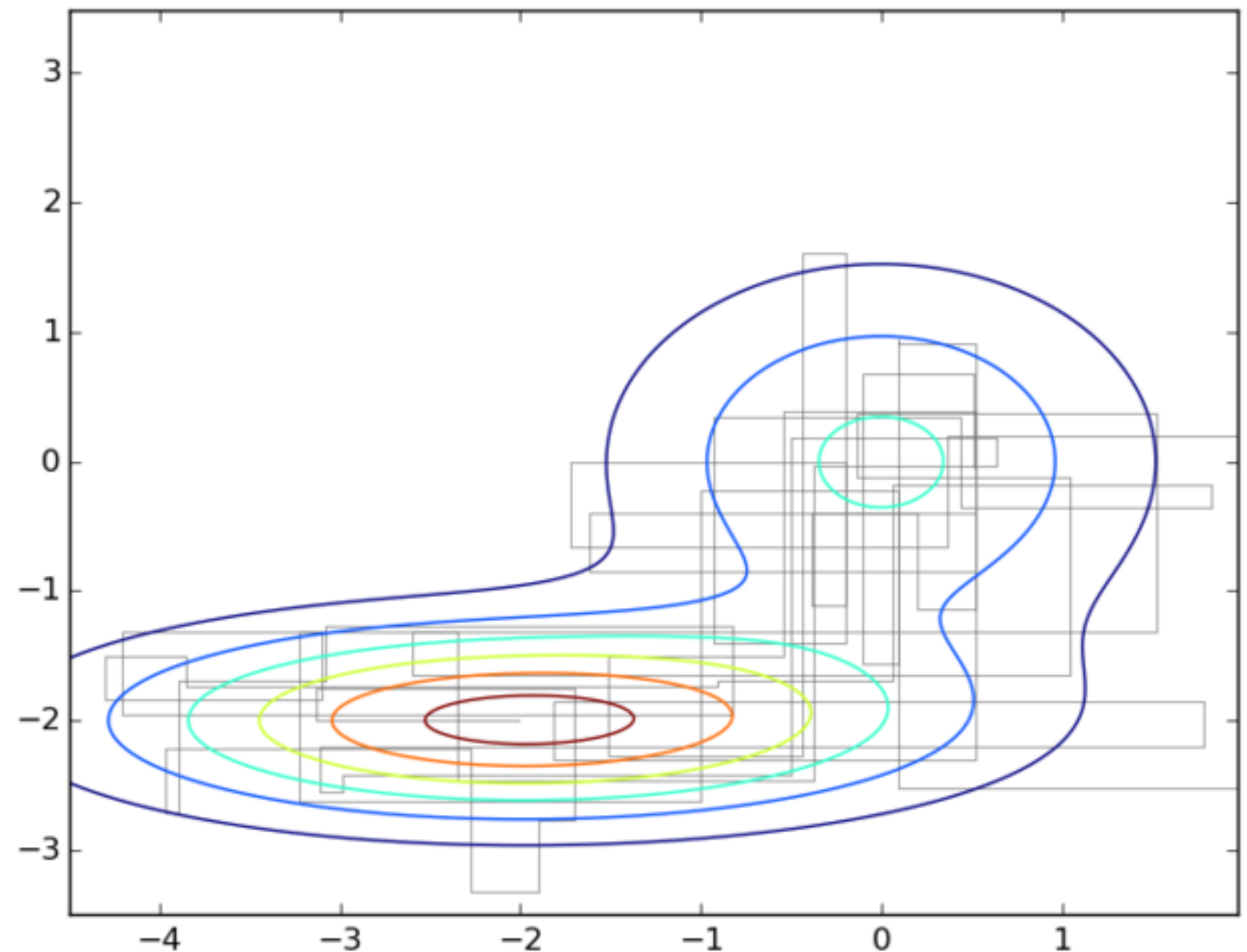
$\Theta, \Phi$  : the matrix of document – topic matrix and the word – topic

Распределение вероятностей слов и документов можно оценить с помощью мат. ожиданий, если знать подинтегральные функции.

$$P(z_i = j | w_i = m, z_{-i}, w_{-i}) \approx \frac{C_{m,j}^{WT} + \beta}{\sum_{m'} C_{m',j}^{WT} + V\beta} \cdot \frac{C_{d,j}^{DT} + \alpha}{C_{d,j'}^{DT} + \alpha T}$$

## Gibbs sampling algorithm

Идея of Gibbs sampling довольно проста: предположим, что мы находимся в очень большом измерении, вектор  $x$  очень велик, и нам сложно выбрать всю выборку сразу (то есть по всем осям), это не получится. Попробуем выбрать выборку не всю сразу, а по компонентно. Это особенно удобно, если каждая компонента независима друг от друга, то есть многомерный вектор - это произведение множества одномерных функций.



$$p(x_i | x_1^{t+1}, \dots, x_{i-1}^{t+1}, x_{i+1}^t, \dots, x_n^t)$$



## Gibbs sampling algorithm

**Input:** коллекция документов  $D$ , количество тем  $|T|$ ,  $L$  - количество итераций, коэффициенты  $\beta$ ,  $\alpha$ ,  $N$  - количество уникальных слов;  
Инициализация:  $\phi(w,t)=1/N$ ,  $\theta(t,d)=1/T$ ;

**Внешний цикл по документам (i)**

**Внутренний цикл по словам в текущем документе.**

Длина цикла  $i$ .

Мы проходим его слово за словом.

Для каждого слова выберите тему

номер в соответствии с распределением:

$$p(w_i) = \frac{C_{mj}^{wt} + \beta}{\sum_m C_{mj}^{wt} + V\beta} \frac{C_{dj}^{dt} + \alpha}{\sum_m C_{dj}^{dt} + \alpha T}$$

Обновляем счетчики

**Конец цикла по словам.**

**Конец цикла по документам**

При наличии счетчиков

Считаем итоговые матрицы:

$$\theta_{dj} = \frac{C_{d,j}^{DT} + \alpha}{C_{d,j}^{DT} + T\alpha}$$

$$\phi_{m,j} = \frac{C_{m,j}^{WT} + \beta}{\sum_{m'} C_{m',j}^{WT} + V\beta}$$



# Brief description of the possibilities and limitations of topic modeling

## Advantages:

1. - один из немногих алгоритмов, который может работать с миллионами документов.
2. ТМ не зависит от языка.
3. Пользователь может работать только с наиболее вероятным распределением слов и документов по темам.
4. Хорошо работает на больших текстах.

## Disadvantages:

1. Существует проблема выбора количества тем (проблема, присущая всем моделям кластеризации). Эта проблема еще не решена до конца.
2. Мы имеем определенную нестабильность. На данный момент предложено несколько схем, которые снижают проблему нестабильности.
3. ТМ плохо работает с короткими текстами. Предложены некоторые решения этой проблемы, но.....есть пространство для исследований



## The problem of stability of TM.

## The ambiguity of the matrix factorization

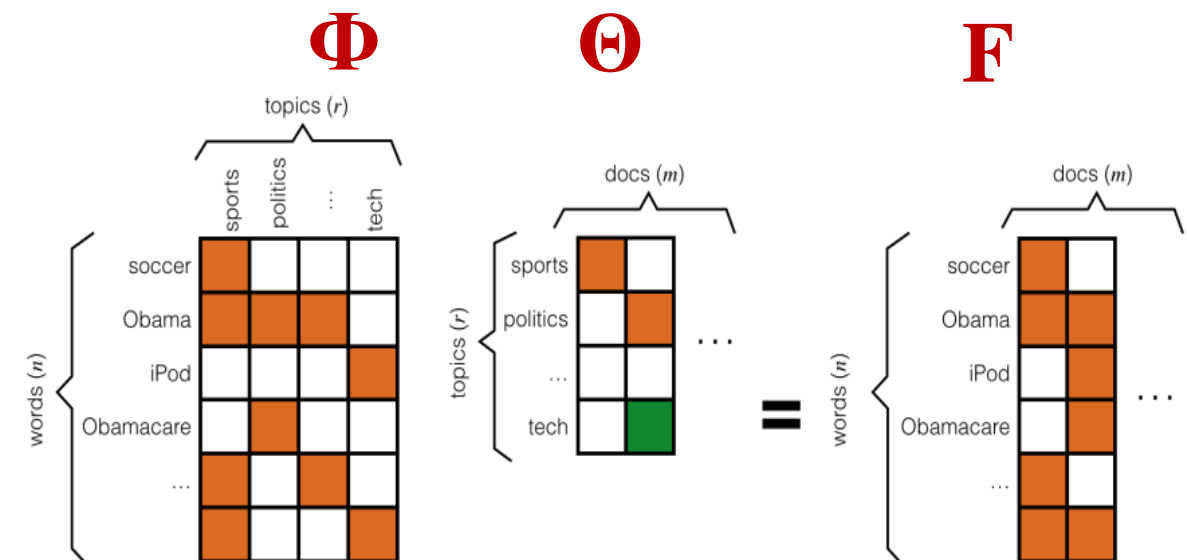
$$F[\text{documents} \times \text{words}] = \Theta[\text{documents} \times \text{topics}] \cdot \Phi[\text{topics} \times \text{words}]$$

Матрица  $F$  набора данных (строки - документы, столбцы - список уникальных слов). Большой набор данных можно представить как произведение двух относительно небольших матриц  $\Phi$  и  $\Theta$ . Однако:

$$F = \Theta \cdot \Phi = (\Theta \cdot R) \cdot (R^{-1} \Phi) = \Theta' \cdot \Phi'$$

Матрица может быть представлена как комбинация различных матриц, но одинакового размера.

Это означает, что исходный набор слов и документов можно сравнивать с разными тематическими решениями, но с одинаковым количеством тем. Содержимое матриц и может отличаться для разных матриц преобразования.

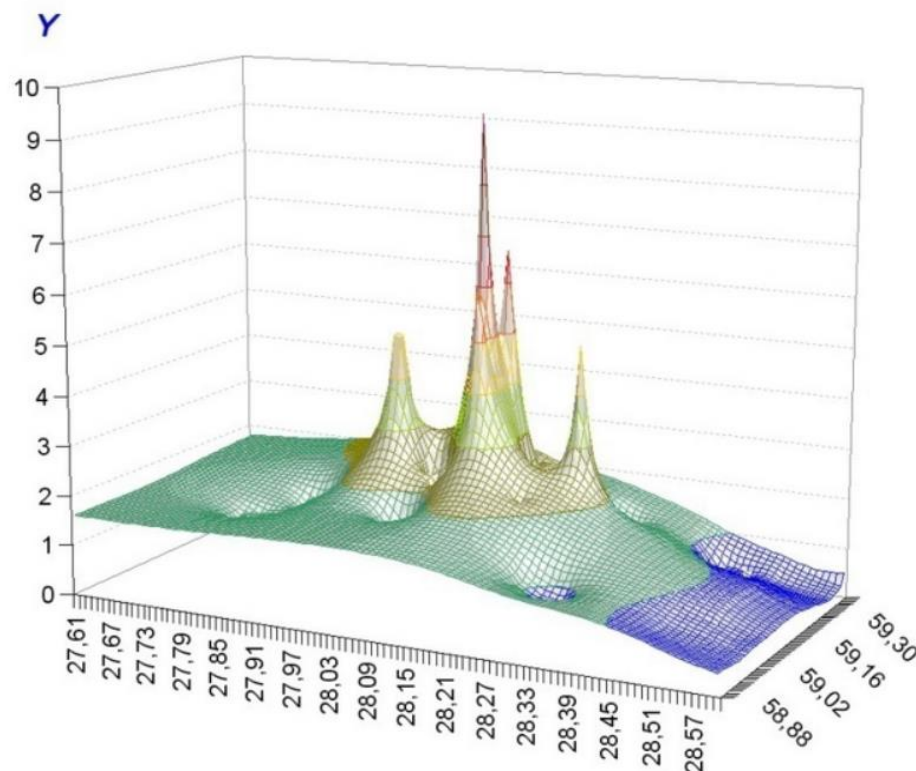


## The problem of stability of TM. Set of local minima and maxima

$$\theta_{m,k} = \int p(t | d) p(\theta_d, \alpha) = \frac{n(k; m) + \alpha_k}{\left( \sum_{k=1}^K \Omega(d; k) + \alpha_k \right)}$$

$$\phi_{k,t} = \frac{n(t; k) + \beta_t}{\left( \sum_{t'=1}^V n(t'; k) + 1 + \beta_{t'} \right)}$$

Интеграл представляет собой произведение функций Дирихле. Полученное подынтегральное выражение имеет набор локальных максимумов и минимумов.

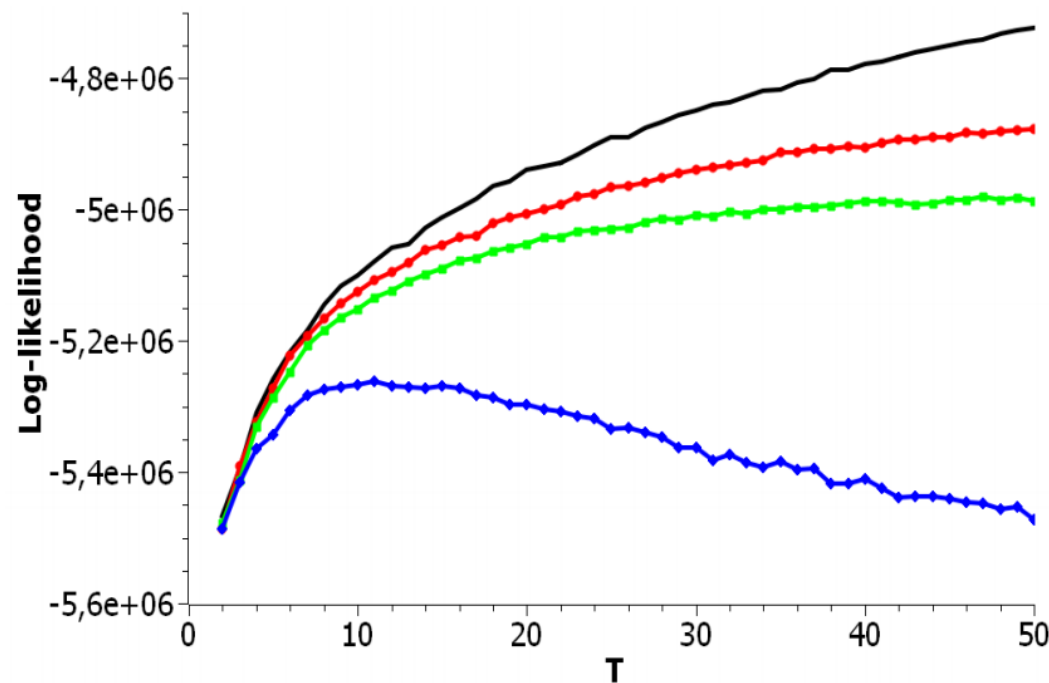


Поскольку вычисление интегралов основано на методе Монте-Карло и сводится к подсчету счетчиков, в результате значения счетчиков могут существенно отличаться при разных запусках процедуры выборки. Это связано с тем, что не все минимумы могут быть обойдены при выборке, и процесс выборки может привести к тому, что итоговая матрица распределений слов и документов по темам будет отражать какой-то один (или несколько) минимум.

## How to determine optimal LDA parameters

logarithm of likelihood – логарифм правдоподобия

$$\ln(P(\hat{D}|\Phi, \Theta)) = \sum_{d=1}^D \sum_{w=1}^W n_d^w \ln\left(\sum_{t=1}^T \phi_{wt} \theta_{td}\right).$$



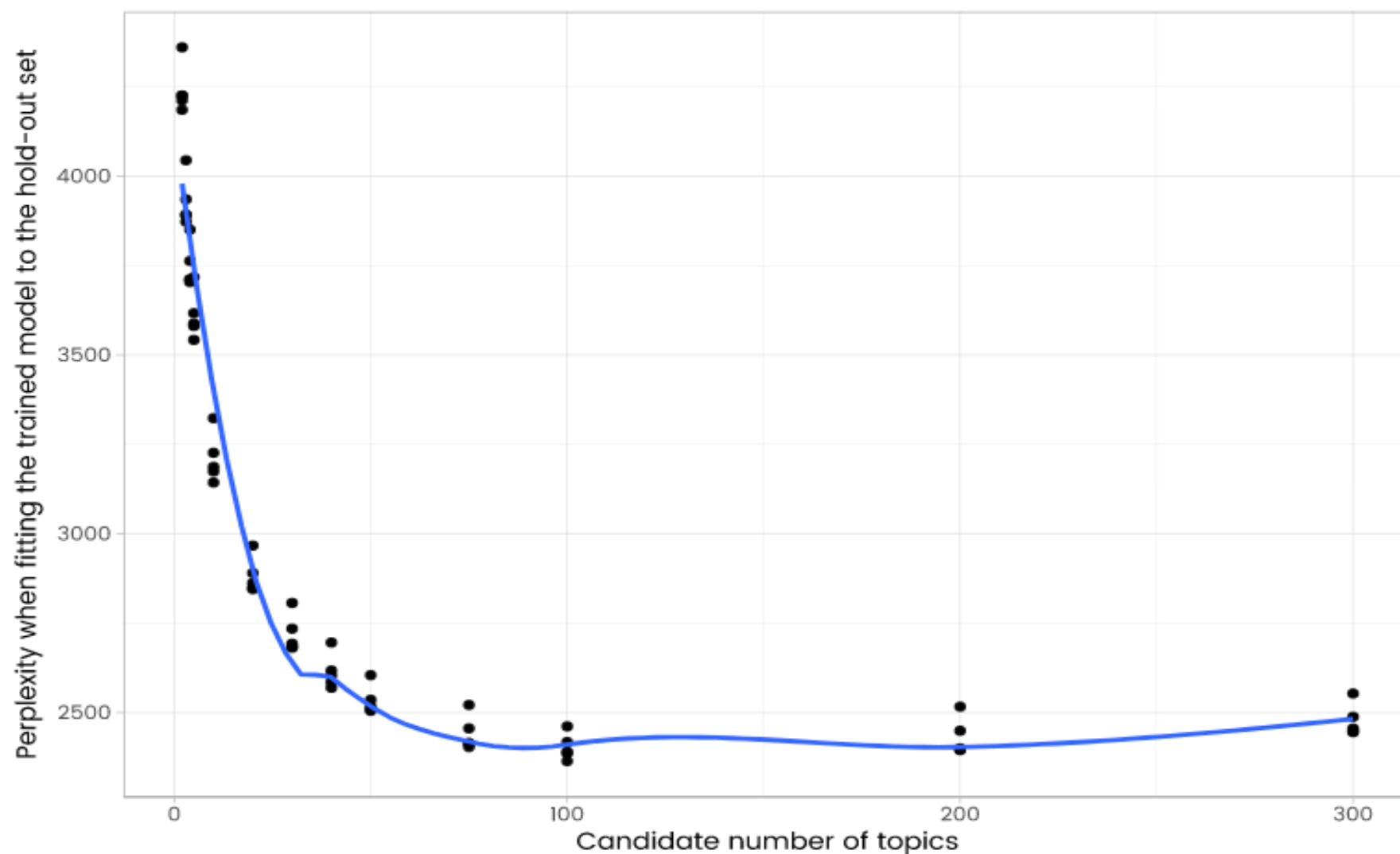
The greater the  
value of likelihood  
is the better

Figure Log-likelihood distribution over  $T$  for different  $\alpha$  and  $\beta$  (Russian dataset).  
pLSA - black, LDA GS ( $\alpha=0.1$ ,  $\beta=0.1$ ) - red, LDA GS ( $\alpha=0.5$ ,  $\beta=0.1$ ) - green, LDA GS ( $\alpha=1$ ,  $\beta=1$ ) - blue.

# How to determine optimal LDA parameters

## Perplexity

$$(D_{\text{test}}) = \exp \left( -\frac{\sum_{d=1}^M \log p(d)}{\sum_{d=1}^M N_d} \right) = \exp \left( -\frac{\sum_{d=1}^M \sum_{w=1}^W n_d^w \log(\sum_{t=1}^T \phi_{wt} \theta_{td})}{\sum_{d=1}^M N_d} \right)$$



The smaller the  
value is the  
better result

## How to measure stability of topic models

**Kullback — Leibler Distance :**

$$K = 0.5 \sum_k^W \Phi_k^1 \log \left( \frac{\Phi_k^1}{\Phi_k^2} \right) + 0.5 \sum_k^W \Phi_k^2 \log \left( \frac{\Phi_k^2}{\Phi_k^1} \right)$$

где  $\Phi^1$  - первое распределение, а  $\Phi^2$  - второе распределение. Если  $K=0$ , то два распределения идентичны, если  $K=\max$ , то два распределения не идентичны. Нормализованная мера КЛ:

$$Kn = \left( 1 - \frac{K}{Max} \right) * 100$$

	A	B	C	D	E	F	G	H
1		Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	Topic 7
2	Topic 1	74.282	63.838	59.917	48.993	53.586	76.325	73.314
3	Topic 2	88.674	80.838	70.908	71.478	76.009	87.92	86.829
4	Topic 3	84.176	75.395	64.271	64.999	72.031	82.857	81.664
5	Topic 4	88.728	81.018	70.969	71.508	76.271	87.986	87.146
6	Topic 5	88.581	81.581	70.435	70.688	75.922	87.739	87.144
7	Topic 6	88.468	80.605	68.769	70.022	79.408	87.373	86.567
8	Topic 7	88.441	80.438	70.388	71.674	76.302	87.847	86.907
9	Topic 8	85.94	77.203	63.889	65.844	73.586	84.55	86.108
10	Topic 9	88.482	80.499	70.156	70.169	76.559	87.823	86.842
11	Topic 10	88.601	80.81	70.185	72.039	76.314	88.216	87.257
12	Topic 11	88.475	80.682	70.065	70.508	76.762	87.573	86.588
13	Topic 12	88.517	80.726	70.33	71.037	76.116	87.639	86.731
14	Topic 13	88.575	80.797	70.158	71.163	76.636	87.574	86.684



## How to measure stability of topic models

$$Kn = \left(1 - \frac{K}{Max}\right) * 100$$

**Level 90 - 93% (and more) means that first 50 words are almost identical.**

**Level about 85%: topics are completely different.**

Similarity 0.935

USA	0.04734	USA	0.03567
American	0.02406	American	0.01804
Syria	0.02082	Syria	0.01758
Obama	0.01374	country	0.01495
weapon	0.01343	war	0.01361
war	0.01309	military	0.01246
president	0.01169	weapon	0.01084
UN	0.01018	Russia	0.01004
military	0.01014	Obama	0.00996
country	0.01005	president	0.0096
chemical	0.00944	UN	0.00869
Syrian	0.00851	international	0.00769

Similarity 0.854

USA	0.04734	water	0.01758
American	0.02406	help	0.01296
Syria	0.02082	city	0.01262
Obama	0.01374	far	0.01199
weapon	0.01343	house	0.01064
war	0.01309	east	0.0104
president	0.01169	region	0.00945
UN	0.01018	dam	0.0091
military	0.01014	flood	0.00904
country	0.01005	resident	0.00839
chemical	0.00944	injured	0.00714
Syrian	0.00851	FRS	0.00698



## Coherence

Topic models изучают темы - обычно представленные как наборы важных слов - автоматически из немаркированных документов без контроля. Это привлекательный метод придания структуры неструктурированным текстовым данным, однако не гарантируется, что темы будут хорошо интерпретируемыми, поэтому были предложены меры согласованности для различения хороших и плохих тем.

**Идея:** тема интерпретируется, если ее главные слова часто встречаются в одном и том же контексте:

$D(v, v')$  be *co-document frequency* of word types  $v$  and  $v'$  (i.e., the number of documents containing one or more tokens of type  $v$  and at least one token of type  $v'$ ), we define *topic coherence* as

$$C(t; V^{(t)}) = \sum_{m=2}^M \sum_{l=1}^{m-1} \log \frac{D(v_m^{(t)}, v_l^{(t)}) + 1}{D(v_l^{(t)})}, \quad (1)$$

where  $V^{(t)} = (v_1^{(t)}, \dots, v_M^{(t)})$  is a list of the  $M$  most probable words in topic  $t$ . A smoothing count of 1 is included to avoid taking the logarithm of zero.



## An overview of the implementations of topic models

<b><u>C/C++/C#</u></b> lda-c hllda	David Blei (variational) Hierarchical LDA, Blei <a href="http://www.cs.columbia.edu/~blei/topicmodeling_software.html">http://www.cs.columbia.edu/~blei/topicmodeling_software.html</a>	<b><u>R/R studio</u></b> Topicmodels lda	Bettina Grün, C Many LDA models Jonathan Chang Collapsed Gibbs sampling (greedy)
<b><u>Java</u></b> Mallet	Andrew McCallum, NLP Toolkit <a href="http://mallet.cs.umass.edu">http://mallet.cs.umass.edu</a>	<b><u>Python</u></b> Gensim, sklearn, tomotopy	Rehurek & Sojka Online LDA (greedy) <a href="https://radimrehurek.com/gensim/about.html">https://radimrehurek.com/gensim/about.html</a>
<b><u>Matlab</u></b> MTMT (Matlab topic modeling toolkit)	Marc Steyvers (Gibbs sampling, C-matlab) <a href="http://psiexp.ss.uci.edu/research/programs_data/toolbox.htm">http://psiexp.ss.uci.edu/research/programs_data/toolbox.htm</a>	<b><u>C+Delphi</u></b> TopicMiner (GUI!)	Sergei Koltcov Topic modeling, preprocessing & testing toolkit <a href="https://linis.hse.ru/en/soft-linis/">https://linis.hse.ru/en/soft-linis/</a>

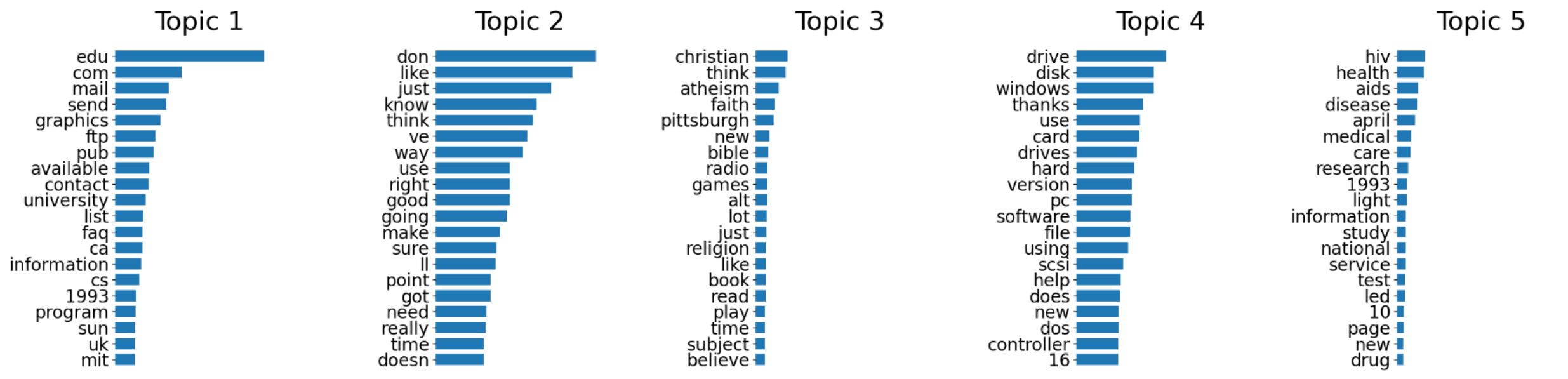


# Implementation of Topic Models in python - Sklearn

## sklearn.decomposition.LatentDirichletAllocation

- Non-negative matrix factorization (NMF)
- Latent dirichlet allocation (LDA)
- TruncatedSVD (also known as latent semantic analysis when used with count or tfidf matrices)

Topics in LDA model





# Implementation of Topic Models in python - BigARTM

- Latent dirichlet allocation (LDA) with regularization
- Hierarchical topic modeling
- Multimodal topic modeling

The following regularizers are implemented in the BigARTM library.

1. Smoothing the distribution of terms in topics. Used to highlight background topics that collect the general vocabulary of the language or the general vocabulary of a given collection.
2. Smoothing topic distributions in documents. Used to highlight background words in each document.
3. Sparse distribution of terms in topics. It is used to highlight the lexical cores of subject topics as a relatively small fraction of the words of the dictionary.
4. Sparse topic distributions in documents. Used to highlight a relatively small proportion of subject topics in each document. Decorating the distributions of terms in topics. It is used to increase the variability of the lexical kernels of subject topics.
5. Selection of topics by zeroing the likelihood of a topic in all documents. It is used to deduce insignificant topics from the model. Allows you to optimize the number of topics, starting with an obviously excessive number of topics and gradually removing unnecessary ones.

<https://bigartm.readthedocs.io/en/stable/>





## Implementation of Topic Models in python - Tomotopy

**tomotopy** is a Python extension of tomoto (Topic Modeling Tool) which is a Gibbs-sampling based topic model library written in C++. It utilizes a vectorization of modern CPUs for maximizing speed. The current version of **tomoto** supports several major topic models including

- Latent Dirichlet Allocation (*tomotopy.LDAModel*)
- Labeled LDA (*tomotopy.LLDAModel*)
- Partially Labeled LDA (*tomotopy.PLDAModel*)
- Supervised LDA (*tomotopy.SLDAModel*)
- Dirichlet Multinomial Regression (*tomotopy.DMRModel*)
- Generalized Dirichlet Multinomial Regression (*tomotopy.GDMRModel*)
- Hierarchical Dirichlet Process (*tomotopy.HDPMModel*)
- Hierarchical LDA (*tomotopy.HLDAModel*)
- Multi Grain LDA (*tomotopy.MGLDAModel*)
- Pachinko Allocation (*tomotopy.PAModel*)
- Hierarchical PA (*tomotopy.HPAModel*)
- Correlated Topic Model (*tomotopy.CTModel*)
- Dynamic Topic Model (*tomotopy.DTModel*)
- Pseudo-document based Topic Model (*tomotopy.PTModel*).

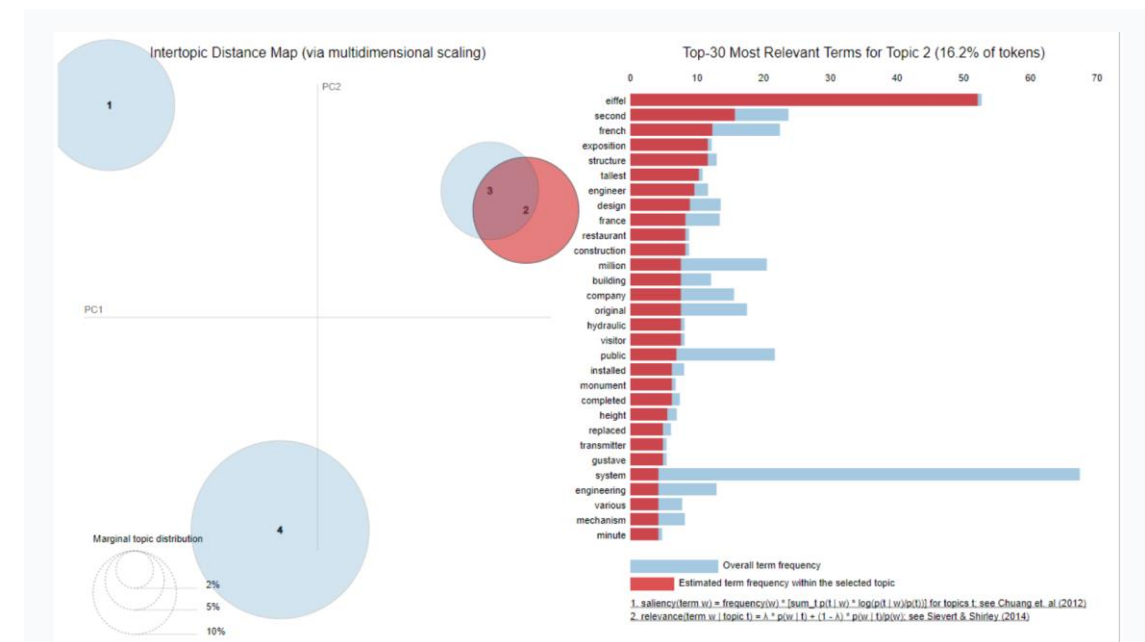


# Implementation of Topic Models in python - Gensim

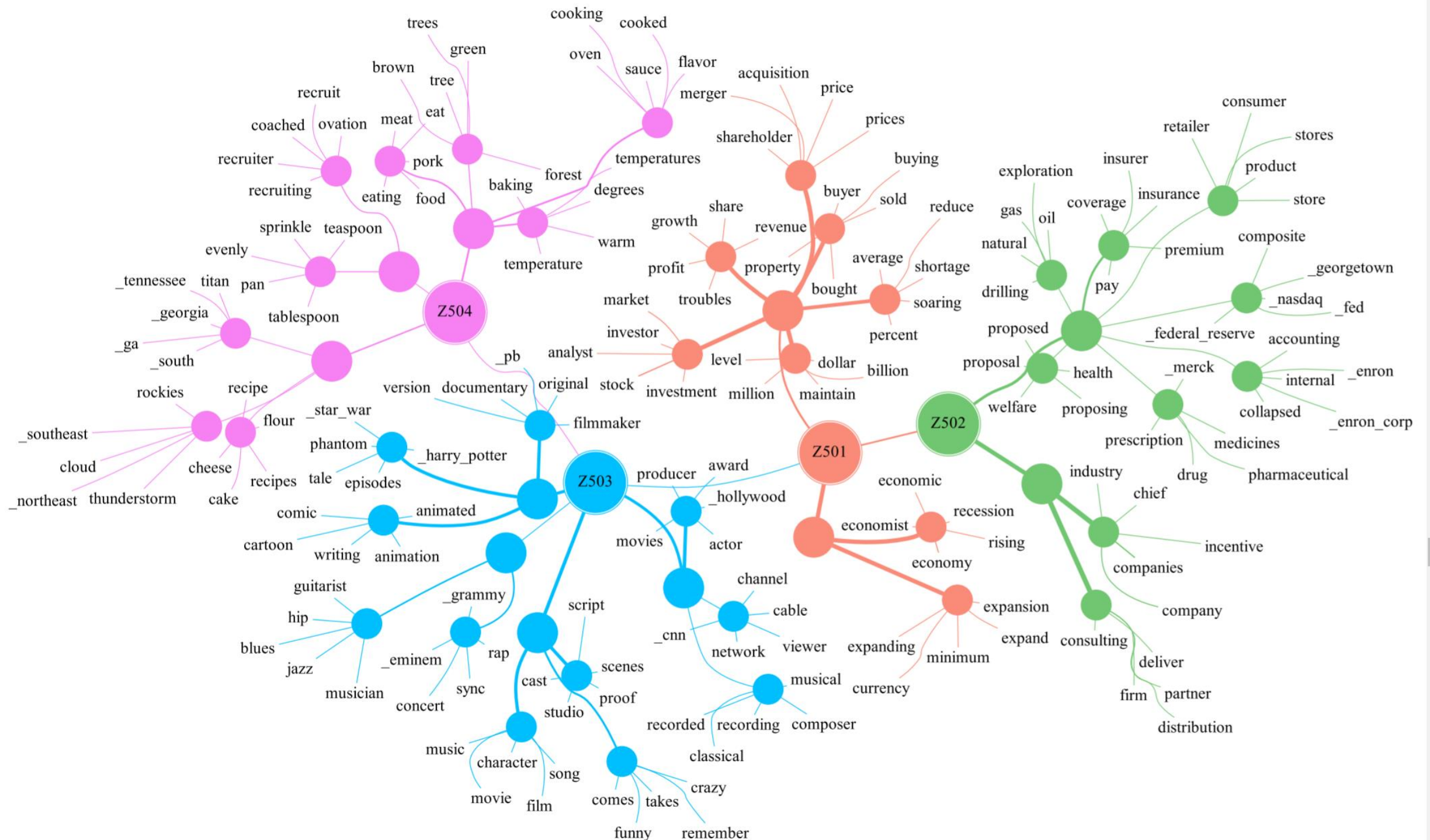
- online Latent Semantic Analysis (LSA/LSI/SVD),
- Latent Dirichlet Allocation (LDA),
- Random Projections (RP),
- Hierarchical Dirichlet Process (HDP)
- word2vec deep learning.

<https://pypi.org/project/gensim/>

Подробное объяснение <https://russianblogs.com/article/56741621473/>  
библиотеки gensim на Python



<https://stackabuse.com/python-for-nlp-working-with-the-gensim-library-part-2/>







# Hierarchical topic modeling

## Модель hLDA

Основная идея непараметрического моделирования заключается в том, чтобы выводить структуру модели из имеющихся данных. Соответственно теоретически, непараметрические тематические модели способны автоматически подбирать число тем по имеющимся данным. Такие непараметрические тематические модели вводят априорное распределение над потенциально бесконечным разбиением целых чисел, используя некоторый стохастический процесс, который присваивает более высокую вероятность решениям с меньшим числом тем.

Модель иерархического скрытого размещения Дирихле (Hierarchical Latent Dirichlet Allocation, hLDA), описанная в работе Блея, основана на вложенном процессе китайского ресторана (Nested Chinese Restaurant Process, nCRP). Chinese Restaurant Process (CRP) генерирует распределение объектов (клиентов) по неограниченному числу разделов (столов). Пусть некоторый китайский ресторан имеет неограниченное (счетное) количество столов.

## Hierarchical topic modeling

В терминах тематического моделирования посетители соответствуют документам, рестораны соответствуют темам. В рамках модели hLDA предполагаются следующие априорные распределения: 1) распределение CRP с параметром  $\gamma$  на возможные деревья; 2) симметричное распределение Дирихле с параметром  $\eta$  на распределение слов по темам ( $\phi_{wt}$ ); 3)  $L$ -мерное распределение Дирихле с параметром  $\alpha$  на пропорции тем в документах ( $\theta_{td}$ ) вдоль пути от корня к листу.

Генеративный процесс модели hLDA описывается следующим образом:

- Для каждого узла  $t$  сэмплируется  $\phi_{.t} \sim Dir(\eta)$
- Для каждого документа  $d$  сэмплируется путь тем  $c_d = \{c_{d1}, \dots, c_{dL}\}$  в соответствии с процессом CRP с параметром  $\gamma$ ; сэмплируются пропорции тем  $\theta_{.d} \sim Dir(\alpha)$ . Для каждой позиции  $n$  слова в документе выбирается уровень  $z_{dn} \sim Mult(\theta_d)$  (как только выбран уровень в пути, тема уже автоматически определена), затем сэмплируется слово из выбранной темы

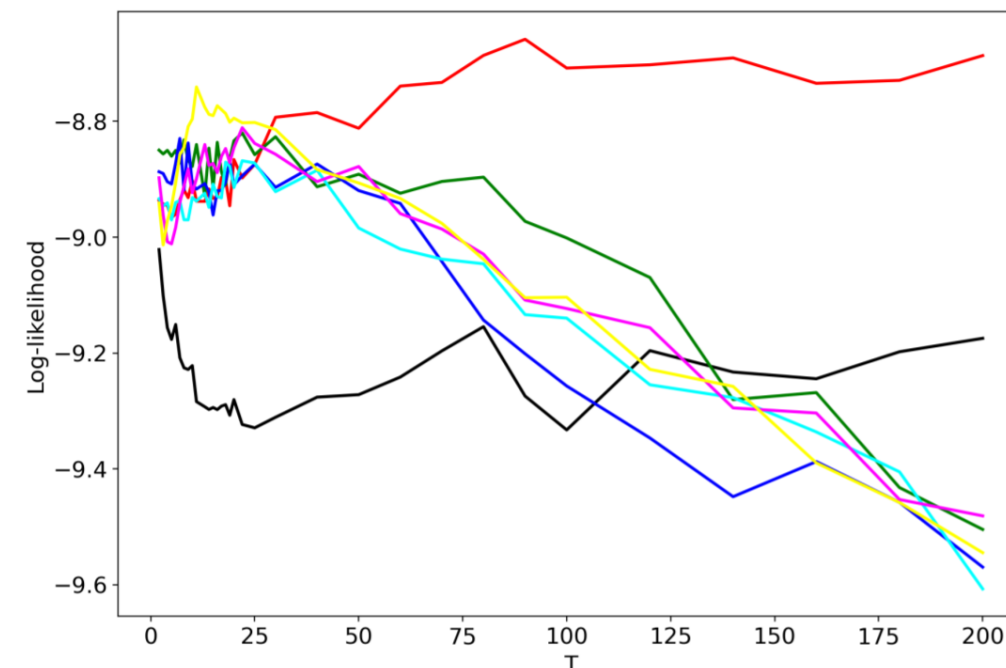
$$w_{dn} \sim Mult(\phi_{.c_d z_{dn}}).$$



# Hierarchical topic modeling

Модель hLDA Отметим, что модель hLDA имеет следующие параметры: 1) глубина иерархии; 2) гиперпараметр  $\alpha$ , который настраивается моделью автоматически; 3) гиперпараметр  $\gamma$ ; 4) гиперпараметр  $\eta$ . Модель hLDA является непараметрической, поэтому она автоматически определяет количество тем на каждом уровне. В данной работе мы исследовали зависимость количества найденных тем от параметра  $\eta$ , который варьировался в диапазоне  $[0.001, 1]$ .

Поскольку данная модель крайне неустойчива и может давать разное количество тем при одинаковых значениях гиперпараметров, мы прогнали модель 10 раз для каждого значения  $\eta$ . Затем мы оценили диапазон полученного количества тем на втором и третьем уровнях.



**Table 3** Range of the derived number of topics by hLDA model for the second ( $T_1$ ) and the third hierarchical levels ( $T_2$ ).

$\eta$	Lenta		20 Newsgroups		Balanced WoS		Balanced Amazon	
	$T_1$	$T_2$	$T_1$	$T_2$	$T_1$	$T_2$	$T_1$	$T_2$
0.001	6–11	31–67	288–358	911–1,402	482–652	1,751–2,242	108–148	561–654
0.01	6–11	13–30	81–111	274–334	68–93	325–453	23–36	108–122
0.2	2–3	5–7	6–11	14–18	2–5	6–13	3	5–6
0.3	2	2–4	4–9	7–11	2–3	3–7	2–3	3–4
0.5	2	2–3	3–5	5–9	2	2–3	2–3	3–4
0.7	2	2–3	3–4	3–7	2	2–3	2	2–4
1	3	2–3	2–4	3–6	2	2–3	2	2–3

# Neural Topic Models

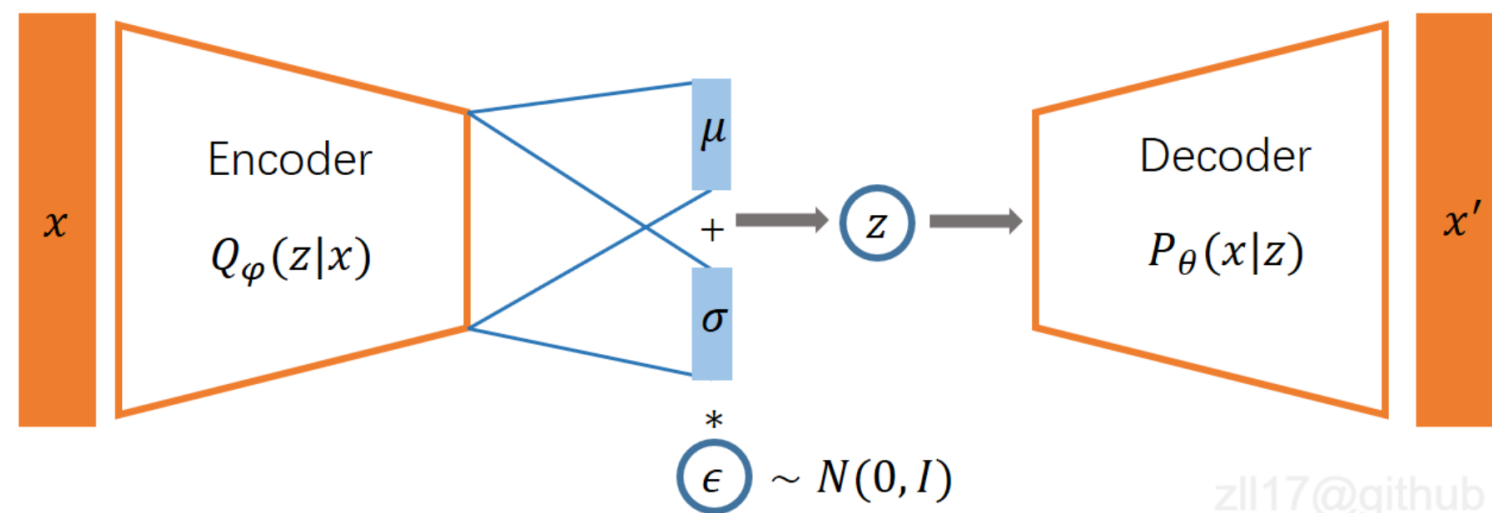
- 1. Installation
- 2. Models
  - 2.1 NVDM-GSM
  - 2.2 WTM-MMD
  - 2.3 WTM-GMM
  - 2.4 ETM
  - 2.5 GMNTM
  - 2.6 BATM
- 3. Datasets
  - 3.1 cnews10k
  - 3.2 zhddline
  - 3.3 zhdd
- 4. Usage
  - 4.1 Preparation
  - 4.2 Run
- 5. Acknowledgement

## 2. Models

### 2.1 NVDM-GSM

Original paper: *Discovering Discrete Latent Topics with Neural Variational Inference*

Author: Yishu Miao





## Example 1. How to use TM in social science

### **Agenda divergence in a developing conflict: Quantitative evidence from Ukrainian and Russian TV newsfeeds**

Media, War & Conflict

1–21

© The Author(s) 2019

Article reuse guidelines:

[sagepub.com/journals-permissions](https://sagepub.com/journals-permissions)

DOI: 10.1177/1750635219829876

[journals.sagepub.com/home/mwc](https://journals.sagepub.com/home/mwc)



**Olessia Koltsova and Sergei Pashakhin** 

National Research University Higher School of Economics, St Petersburg, Russia

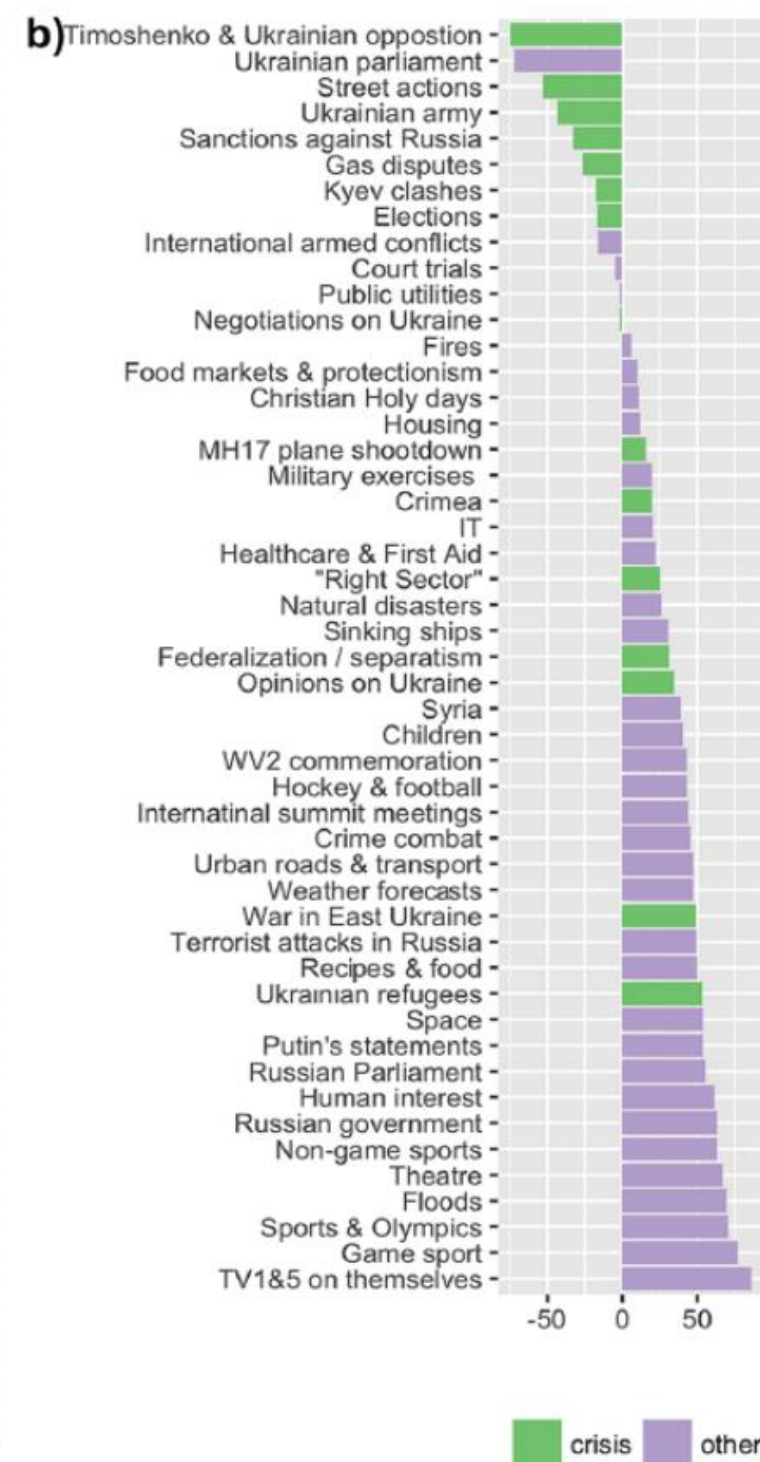
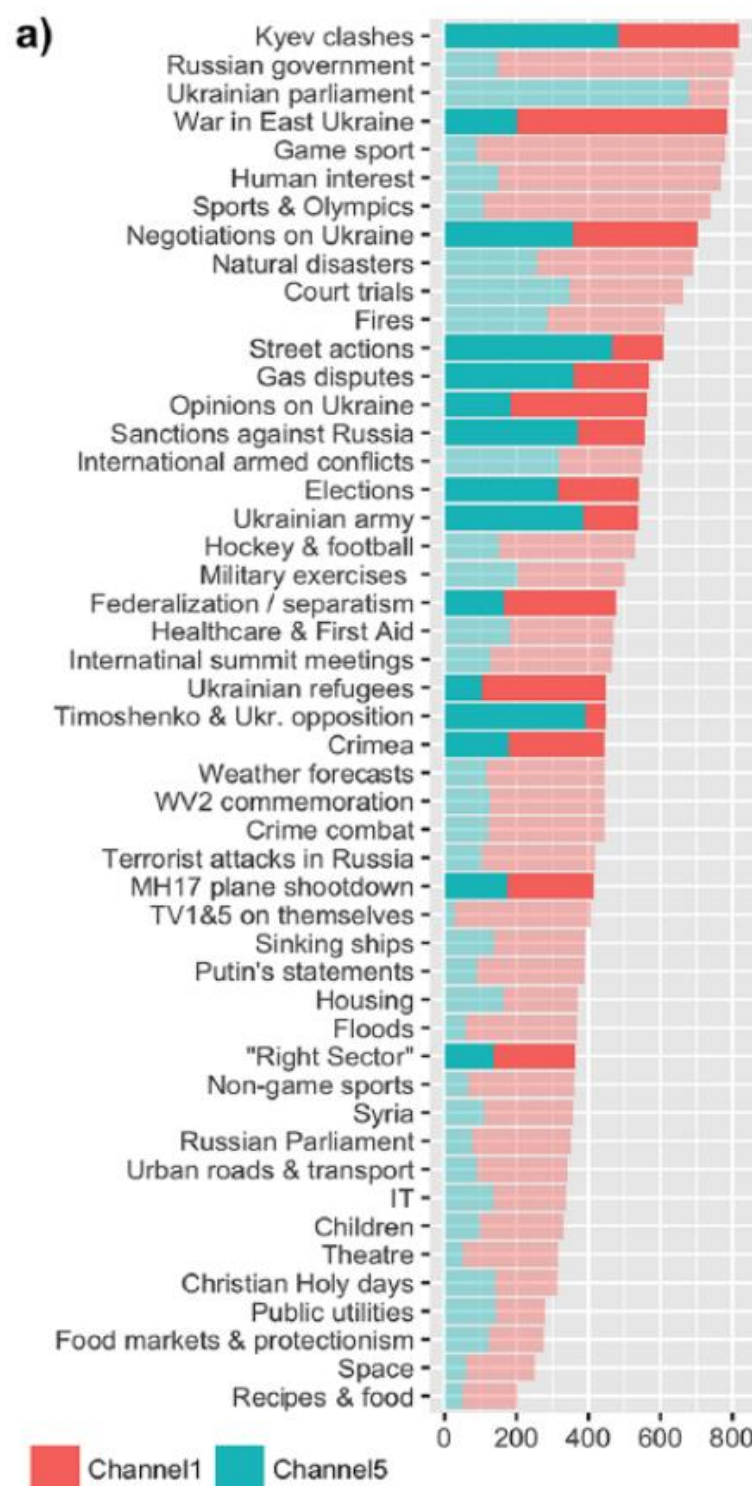
В этой статье авторы представляют количественные свидетельства расхождения повестки дня между СМИ конфликтующих сторон в ходе украинского кризиса 2013–2014 годов. Используя 45 000 сообщений из онлайн-лент новостей российского и украинского телеканалов, они провели тематическое моделирование в сочетании с качественным анализом, чтобы выявить темы, связанные с кризисом, оценить их значимость и составить карту динамики внимания обоих каналов к каждой из этих тем. Выясняется, что оба канала формируют принципиально разные повестки дня. На основе украинского случая они предлагают типологию этапов освещения конфликта в СМИ.



## Example 1. How to use TM in social science

Чтобы отразить в новостях основные события кризиса, авторы выбрали период в 53 недели со 2 сентября 2013 года по 7 сентября 2014 года. Этот период начинается за 11 недель до кризиса, чтобы обеспечить выборку некризисного освещения.


Российское телевидение - 1 канал (официальная позиция государства). Украинский канал - 5 каналов (принадлежит президенту Петру Порошенко)





## Example 2. How to use TM in social science

### A Full-Cycle Methodology for News Topic Modeling and User Feedback Research

Sergei Koltsov<sup>1</sup> , Sergei Pashakhin<sup>1</sup> , and Sofia Dokuka<sup>2</sup> 

<sup>1</sup> National Research University Higher School of Economics,  
St. Petersburg 190008, Russia  
skoltsov@hse.ru

<sup>2</sup> Institute of Education, National Research University Higher School  
of Economics, Moscow 101000, Russia

В данной работе мы предлагаем методологию полного цикла такого исследования: от выбора оптимального количества тем до выделения устойчивых тем и анализа результатов ТМ. Авторы иллюстрирует ее на примере анализа новостного онлайн-потока из 164.426 сообщений, сформированного двенадцатью национальными телеканалами за годовой период (данные из ВК). Авторы показывают, что метод позволяет легко выявить ассоциации между темами новостей и отзывами пользователей, в том числе их поведением при совместном использовании. Кроме того, авторы показывает, как неравномерное распределение количества и длины документов по классам (телеканалам) может повлиять на результаты ТМ.





## Example 2. How to use TM in social science

Channel proportion in topics

**Table 3.** Contribution of TV channels into topic salience, “likability” and “shareability”

Stable topic	Channel weight in a topic, %	Channel likes, %	Channel reposts, %
Mixture of controversial events	RIA News – 98%	RIA News – 99.54%	RIA News – 99.80%
Russian sport achievements	Russia Today – 41.29%	Russia Today – 31.90%	Russia Today – 26.75%
	Russia-24 – 17.43%	Russia-24 – 8.76%	Russia-24 – 10.96%
	RIA News – 16.6%	RIA News – 47.03%	RIA News – 42.19%
	NTV – 12.45%	NTV – 2.86%	NTV – 4.61%
Syria & Russia	RIA News – 33.68%	RIA News – 54.46%	RIA News – 44.63%
	Russia Today – 30.58%	Russia Today – 33.42%	Russia Today – 36.60%
	Russia-24 – 13.00%	Russia-24 – 4.93%	Russia-24 – 7.05%
	NTV – 10.95%	NTV – 2.40%	NTV – 4.36%
Russian athletes doping controversy	Russia Today – 30.50%	Russia Today – 24.66%	Russia Today – 23.42%
	RBC – 17.3%	RBC – 17.13%	
		RIA News – 37.75%	RBC – 20.15%
		NTV – 5.83%	RIA News – 30.52%

## Example 2. How to use TM in social science

Channel proportion in topics

**Table 3.** *(continued)*

Stable topic	Channel weight in a topic, %	Channel likes, %	Channel reposts, %
Street actions & protests (international)	Dozhd – 25.67%	Dozhd – 12.19%	Dozhd – 15.38%
	RBC – 22.77%	RBC – 40.83%	RBC – 47.90%
	RIA News – 20.08%	RIA News – 33.49%	RIA News – 19.33%
	NTV – 13.04%	NTV – 2.08%	NTV – 2.89%
Putin & his addresses	NTV – 21.85%	NTV – 5.66%	NTV – 8.73%
	Russia Today – 21.65%	Russia Today – 22.88%	Russia Today – 19.08%
			RBC – 23.44%
	RBC – 14.64%	RBC – 13.54%	RIA News – 35.62%
FSB and counterterrorist activities	RIA News – 12.57%	RIA News – 48.68%	
	NTV – 23.81%	NTV – 6.78%	NTV – 9.05%
	RIA News – 20.08%	RIA News – 57.47%	RIA News – 40.94%
	Russia-24 – 13.25%	Russia-24 – 7.45%	Russia-24 – 10.62%
	Russia Today – 12.42%	Russia Today – 12.33%	Russia Today – 14.10%



NATIONAL RESEARCH  
UNIVERSITY

# Thank you for your attention

<https://linis.hse.ru/>

Phone: +7 (911) 981 9165

Email: [skoltsov@hse.ru](mailto:skoltsov@hse.ru)