Multimodal topic modeling for exploratory search in collective blog

Anastasia Yanina • yanina-n@yandex-team.ru
Konstantin Vorontsov • voron@forecsys.ru

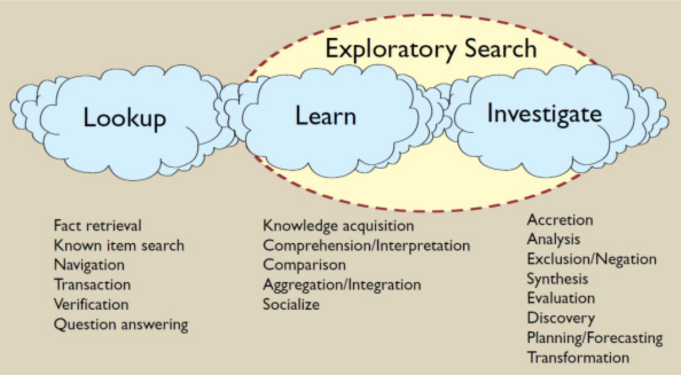
11th International Conference on Intelligent Data Processing: Theory and Applications

IDP-2016

BARCELONA

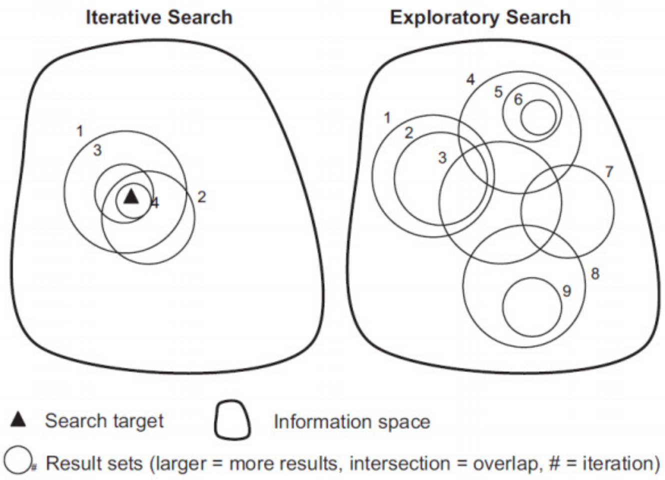
The paradigm of Exploratory Search

- what if the user doesn't know which keywords to use?
- what if the user isn't looking for a single answer?



Gary Marchionini. Exploratory Search: from finding to understanding. Communications of the ACM. 2006, 49(4), p. 41–46.

Iterative "query-browse-refine" search vs Exploratory Search



R.W.White, R.A.Roth. Exploratory Search: beyond the Query-Response paradigm. San Rafael, CA: Morgan and Claypool, 2009.

Exploratory search

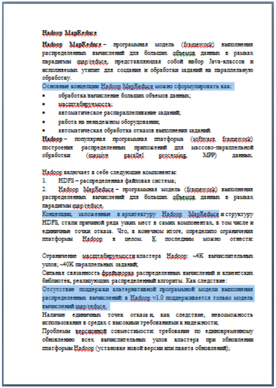
Query

Exploratory query is a description of user's search intention (1-2 pages of text)

Search results

Result of exploratory search is a set of relevant articles.

A user should be able to create a complete picture of the subject area after looking through the search results.



Hadoop MapReduce

Hadoop MapReduce – программная модель (framework) выполнения распределенных вычислений для больших объемов данных в рамках парадигмы map/reduce, представляющая собой набор Java-классов и исполняемых утилит для создания и обработки заданий на параллельную обработку.

Основные концепции Hadoop MapReduce можно сформулировать как:

- обработка/вычисление больших объемов данных;
- масштабируемость;
- автоматическое распараллеливание заданий;
- работа на ненадежном оборудовании;
- автоматическая обработка отказов выполнения заданий.

Hadoop – популярная программная платформа (software framework) построения распределенных приложений для массово-параллельной обработки (massive parallel processing, MPP) данных.

Hadoop включает в себе следующие компоненты:

1. HDFS – распределенная файловая система;
2. Hadoop MapReduce – программная модель (framework) выполнения распределенных вычислений для больших объемов данных в рамках парадигмы map/reduce.

Концепции, заложенные в архитектуру Hadoop MapReduce и структуру HDFS, стали причиной ряда узких мест в самих компонентах, в том числе и единичные точки отказа. Что, в конечном итоге, определило ограничения платформы Hadoop в целом. К последним можно отнести:

Ограничение масштабируемости кластера Hadoop: ~4K вычислительных узлов; ~40K параллельных заданий;

Сильная связанность фреймворка распределенных вычислений и клиентских библиотек, реализующих распределенный алгоритм. Как следствие:

Отсутствие поддержки альтернативной программной модели выполнения распределенных вычислений: в Hadoop v1.0 поддерживается только модель вычислений map/reduce.

Наличие единичных точек отказа и, как следствие, невозможность использования в средах с высокими требованиями к надежности;

Проблемы версионной совместимости: требование по единовременному обновлению всех вычислительных узлов кластера при обновлении платформы Hadoop (установке новой версии или пакета обновлений);

Example of query for exploratory search

Multimodal topic model

D — set of documents (collective blog articles)
T — set of topics,
M — set of modalities,
$W^1, \dots, W^m$ — dictionaries for each modality $m \in M$.

Modalities: words, authors, comment authors, tags, categories.

Φ **matrix of term distributions of topics** for modality m:

$$\Phi_m = (\phi^m_{wt})_{W^m \times T} \qquad \phi^m_{wt} = p(w|t) \quad \forall m \in M$$

Θ **matrix of topic distributions of documents:**

$$\Theta = (\theta_{td})_{T \times D}, \qquad \theta_{td} = p(t|d)$$

Multimodal ARTM (Additively Regularized Topic Model)

Maximum log-likelihood with multiple modalities and regularization:

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

where $R(\Phi, \Theta) = \sum_{i=1}^{n} \tau_i R_i(\Phi, \Theta)$ is a combination of regularizers.

EM-algorithm is a simple iteration method for the system

$$\begin{array}{rl} \text{E-step:} & \left\{ \begin{array}{l} p_{tdw} = \operatorname*{norm}_{t\in T}\big(\phi_{wt}\theta_{td}\big) \\[2ex] \phi_{wt} = \operatorname*{norm}_{w\in W^m}\Big(\sum_{d\in D} \lambda_{m(w)} n_{dw} p_{tdw} + \phi_{wt}\frac{\partial R}{\partial \phi_{wt}}\Big) \\[2ex] \theta_{td} = \operatorname*{norm}_{t\in T}\Big(\sum_{w\in d} \lambda_{m(w)} n_{dw} p_{tdw} + \theta_{td}\frac{\partial R}{\partial \theta_{td}}\Big) \end{array} \right. \\ \text{M-step:} & \end{array}$$

BigARTM project

BigARTM features:

- Parallel + Online + Multimodal + Regularized Topic Modeling
- Out-of-core one-pass processing of large text collection
- Built-in library of regularizers and quality measures

BigARTM community:

- Open-source https://github.com/bigartm
  (discussion group, issue tracker, pull requests)
- Documentation http://bigartm.org

BigARTM license and programming environment:

- Freely available for commercial usage (BSD 3-Clause license)
- Cross-platform — Windows, Linux, Mac OS X (32 bit, 64 bit)
- Programming APIs: command-line, C++, and Python

Data from collective blog habrahabr.ru

Data

- 132 157 articles (in Russian)
- Metadata:
  - author
  - tags and categories
  - comments and their authors
  - number of article views
  - number of article likes

Modalities of the collective blog

- Terms: 52354 unigram words
- Article authors: 1000 users
- Comment authors: 10000 users
- Tags: 2546
- Categories: 123

Regularizers and quality criteria

Regularizers

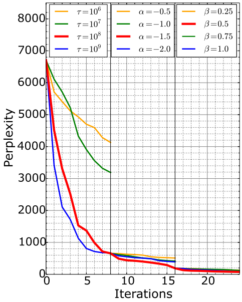
- Decorrelation for terms in topics
- Smoothing for terms in topics
- Sparsity of topics in documents
- Background topics to highlight common vocabulary words

Quality criteria

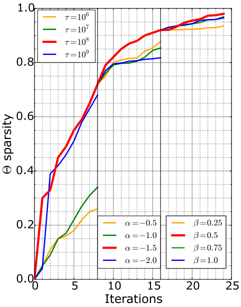
- Perplexity
- Sparsity of terms in topics
- Sparsity of topics in documents

Greedy coordinate-wise multicriteria optimization of regularization coefficients

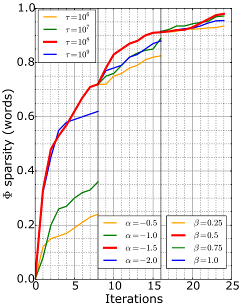
We add regularizers one by one to improve sparsity without loss of the perplexity.



Perplexity



Θ sparsity



Φ sparsity (words)

Topical exploratory search

1. Learn a topic model from a text collection (offline)
2. Calculate a topic representation of the query (quick online)
3. Rank documents by topical similarity to the query
4. Use top k documents as search result

$q = (w_1, \dots, w_{n_q})$ — query text of n_q terms

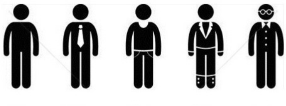
$\theta_{tq} = p(t|q)$ — topic distribution of query q
$\theta_{td} = p(t|d)$ — topic distribution of document $d \in D$

Cosine measure of similarity between document d and query q:

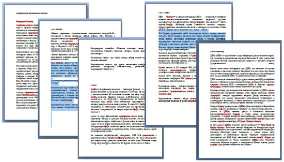
$$\mathrm{sim}(q, d) = \frac{\sum_t \theta_{tq}\theta_{td}}{\left(\sum_t \theta_{tq}^2\right)^{1/2}\left(\sum_t \theta_{td}^2\right)^{1/2}}.$$

Inverted index can by used for search documents d by query topics t

Evaluation of the exploratory search quality



Assessors



Queries

Two tasks for assessors:

1. Find as much as possible relevant articles using any tools (search engines, searching by tags, etc.)
2. Evaluate the relevance of topical search for the same query.

Examples of ES-query titles in our experiment

Algorithms for coloring graphs	IBM Watson
Netflix	3D-printing
Techniques for fast typing	CERN-кластер
Elon Mask space projects	AB-testing
Hadoop MapReduce	Cloud computing services
Self-driving Google car	Contextual advertising
Public-key cryptography	Rover Curiosity
Platforms for online education	Videocards NVIDIA
Data Science Meetups in Moscow	Pattern recognition
Educational projects mail.ru	Google scholar
Interplanetary station New horizons	MIT MediaLab Research
Word2vec	Microsoft Azure

Results of search quality evaluation

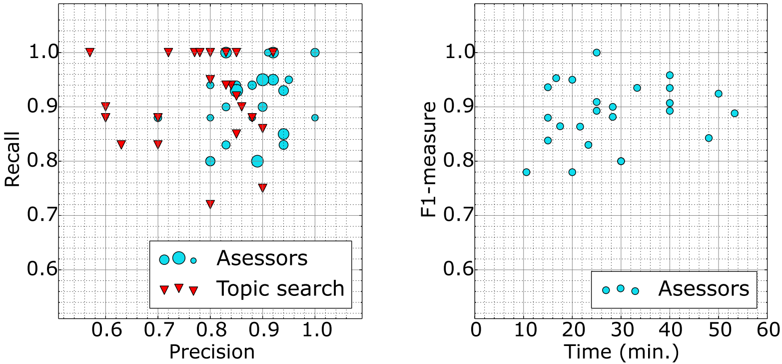
Number of queries: 25 (10 are shown in the table, averages by 25)
Number of assessors per query: 3
Average time for processing query: 30 minutes

Automatic topical search vs. assessors' search

Assessors				**Topical search**		
search time	docs found	Preci-sion	Recall	docs found	Preci-sion	Recall
48	9	0.89	0.80	12	0.83	**1.0**
40	25	0.92	0.95	25	0.92	**1.0**
15	10	0.80	0.88	11	0.72	**1.0**
40	18	0.94	**0.85**	20	0.85	**0.85**
40	55	0.92	**1.0**	57	0.84	0.94
15	12	0.91	**1.0**	14	0.57	**1.0**
25	12	0.94	**0.83**	10	0.90	0.75
28	12	0.83	**0.9**	10	0.80	0.72
50	7	0.88	**0.88**	10	0.70	**0.88**
45	15	0.94	**0.93**	23	0.60	0.88
average:	**18**	**0.87**	**0.89**	**20**	**0.77**	**0.91**

Results of search quality evaluation

Assessors vs. topical search: Precision, Recall, F1, Time



Precision and Recall

Time and f-measure

Results of search quality evaluation (in average)

Number of queries: 25 (10 are shown in the table, averages by 25)
Number of assessors per query: 3
Average time for processing query: 30 minutes

Automatic topical search vs. assessors' search (all metrics are averaged by queries)

Metric	assessors	topical search
Precision@5	0.82	0.74
Precision@10	0.87	0.77
Precision@15	0.86	0.68
Precision@20	0.85	0.68
Recall@5	0.78	**0.82**
Recall@10	0.84	**0.88**
Recall@15	0.88	**0.90**
Recall@20	0.88	**0.91**

Finding the optimal number of topics in model

The advantage of our evaluation technique:

Asking assessors once, we can evaluate and compare many models

Assessors' vs. topical search: Precision@k and Recall@k,
for the model with 5 modalities and different number of topics $|T|$

	asessors	100	**200**	300	400	500
Precision@5	0.82	0.61	**0.74**	0.71	0.69	0.59
Precision@10	0.87	0.65	**0.77**	0.72	0.67	0.61
Precision@15	0.86	0.67	**0.68**	0.67	0.65	0.62
Precision@20	0.85	0.64	**0.68**	0.67	0.64	0.60
Recall@5	0.78	0.62	**0.82**	0.80	0.72	0.63
Recall@10	0.84	0.63	**0.88**	0.81	0.75	0.64
Recall@15	0.88	0.67	**0.90**	0.82	0.77	0.67
Recall@20	0.88	0.69	**0.91**	0.85	0.77	0.68

Finding the optimal set of modalities

The advantage of our evaluation technique:

Asking assessors once, we can evaluate and compare many models

Assessors' vs. topical search: Precision@k and Recall@k,
with fixed $|T| = 200$ and different sets of modalities
(Words, Tags, Hubs (categories), Authors, Comment authors)

	assessors	W	C	TH	WT	WH	WTH	**WTHAC**
Pr@5	0.82	0.63	0.54	0.59	0.74	0.73	0.73	**0.74**
Pr@10	0.87	0.67	0.56	0.58	0.77	0.74	0.75	**0.77**
Pr@15	0.86	0.65	0.53	0.55	0.67	0.67	0.68	**0.68**
Pr@20	0.85	0.64	0.53	0.54	0.66	0.67	0.68	**0.68**
Recall@5	0.78	0.77	0.63	0.69	0.82	0.81	0.82	**0.82**
Recall@10	0.84	0.79	0.64	0.71	0.88	0.82	0.87	**0.88**
Recall@15	0.88	0.82	0.67	0.73	0.90	0.84	0.89	**0.90**
Recall@20	0.88	0.85	0.68	0.74	0.91	0.85	0.89	**0.91**

Conclusions & Contacts

- We used ARTM for the topical Exploratory Search
- We proposed the evaluation technique for Exploratory Search
- The automatic topical Exploratory Search is much faster than assessors' one, having comparable quality

Yanina Anastasia

Analyst, Yandex LLC
Moscow Institute of Physics and Technology

yanina-n@yandex-team.ru