Regularized Multimodal Hierarchical Topic Model for Document-by-Document Exploratory Search

Anastasia Ianina, Konstantin Vorontsov
Moscow Institute of Physics and Technology
Moscow, Russia
yanina, k.v.vorontsov@phystech.edu

***Abstract*—In the exploratory search paradigm of information retrieval, the user has a complicated search demand that can not be formulated in a short query. The user collects thematically relevant information iteratively in a "query–browse–refine" process being motivated by learning, understanding, and knowledge acquisition purposes. We consider an elementary step of this scenario in which the search intent can be expressed by a long text query. For this case, we develop an exploratory search engine based on probabilistic topic modeling. Topic model gives a low-dimensional sparse interpretable vector representation (topical embedding) of a text. The search engine uses these embeddings for ranking documents by their similarity to the query. We show that performing only one query, the topic-based search engine achieves better precision and recall that human assessors do spending up to one hour in a conventional browse–refine loop. We use *additive regularization for topic modeling* (ARTM) to make the model simultaneously sparse, decorrelated, n-gram, multimodal and hierarchical. We show experimentally that each of these features of the model is important to achieve precision and recall higher than 90%. Topical hierarchy emulates a natural human strategy to focus on subtopics gradually discarding unnecessary information. Also we show that increasing the number of levels in the hierarchy improves the search quality and makes it possible to enrich the model with a larger number of topics. We use the fast parallel implementation of the regularized EM-algorithm from BigARTM open source project. We use crowdsourcing in order to collect relevance assessments for the search quality evaluation.**

I. Introduction

The conventional search engines take in short and precisely formulated queries from mass users. Exploratory search is a relatively new paradigm in information retrieval. It aims at self-education, acquisition and systematization of knowledge [15], [27]. The key distinguishing mark of the exploratory search is the absence of the exact query and the unique result. The user of a conventional search system has to formulate many short queries iteratively, gradually expanding the search domain by repeated steps of querying, browsing search results, and refining the query. In such cases it is easier to come up with a broad search direction indicated by a long text query, possibly by a whole document, a fragment of a document, or a set of documents.

Topic modeling is relatively new approach in the literature on exploratory search [20], [9], [18], [24]. The interpretability, the flexibility and the possibility of visualization and navigating are said to be the key advantages of topic models for exploratory search [24]. On the other hand, exploratory search is often said to be one of the key applications in topic modeling literature, and searching for semantically similar documents is often used for model validation [29], [1]. However these studies have not led to the creation of effective freely available solutions for topic-based exploratory search yet. Our work falls into the recent trend to merge these two research directions.

In this work we present a topic-based approach to exploratory search. A probabilistic topic model extracts a set of latent topics from a collection of text documents. It defines each topic by a probability distribution over words and describes each document with a probability distribution over topics [11], [4], [3].

The full text search is usually based on inverted index and looks for documents, which contain all the words from the query [14]. So, if the query is long, it's most likely that nothing will be found. Topic-based search overcomes this problem by comparing low-dimensional topical vector representations of query and documents instead of their bag-of-words representations.

Topic models have to meet a non-trivial combination of requirements in the exploratory search system. They must automatically build significantly different and well interpretable topics; divide topics into subtopics hierarchically; take into account word collocations and meta-data such as authorship, time, categories, tags etc. All these aspects have been elaborated separately in special topic models, mainly within Bayesian learning framework [3], [6]. However, it is technically difficult to aggregate them remaining in Bayesian approach. We bypass this by using a non-Bayesian theory of *additive regularization for topic modeling* (ARTM) [25].

In ARTM a topic model is learned from the collection by maximizing a weighted sum of the log-likelihood and additive regularization criteria. Regularizers can be borrowed from known topic models including Bayesian ones. ARTM encourages a flexible modular way to combine topic models by turning regularizers on or off, thus creating a model with desired properties [26], [2], [12]. The optimization problem in ARTM is solved by a general regularized expectation-maximization (EM) algorithm. We use an effective parallel implementation of the online EM-algorithm from open source project BigARTM.org [8]. Compared with the previous work [28], in this paper we introduce a topical hierarchy and show that it significantly improves both precision and recall. Moreover, increasing the number of levels in the hierarchy improves the search quality and enriches the model with a larger number of topics. Also we compare our model with stronger highly competitive baselines.

The rest of the paper is organized as follows. In sec-

tion II we introduce basic notation and define the additively regularized topic model for exploratory search. In section III we describe a hierarchical topic model for the cascade topic-based search. In section IV we propose a topic-based search algorithm and design the experiment to compare the results of assessors' iterative search and topic-based exploratory search. Note, that there is no iterative query reformulation in the topic-based exploratory search. Hence, we don't need complicated methods to evaluate the user behavior like those used in [13], [17], [21]. In sections V-B and V-C we evaluate the search quality on two popular tech news media: TechCrunch in English and Habrahabr in Russian. We compare topic-based search both with manual search performed by assessors and several baselines including TF-IDF, BM25, word embeddings, CNN-based approaches [10], siamese adaptation of LSTM — MaLSTM [16] and Tree LSTM [22]. In section VI we show how to tune a hierarchy of topics and provide some technical details for reproducing our results. In section VII we conclude and discuss some potentials and limitations of the topic-based exploratory search.

II. Probabilistic topic modeling

Let us consider a collection D of multimodal documents. Each document d from D contains terms of multiple modalities, such as words, bigrams, tags, categories, authors, etc. Each modality m from the finite set of modalities M is defined by term dictionary W_m. Term frequency n_{dw} is the number of times the term w appears in the document d.

Topic modeling is based on the assumption that the appearance of each term in a document can be explained by some latent topic from a given finite set of topics T. *Probabilistic topic model* describes the observable term frequencies in each document by a probabilistic mixture of term distribution for the topics $\phi_{wt} = p(w|t)$ weighted by topic probabilities for the documents $\theta_{td} = p(t|d)$:

$$p(w|d) = \sum_{t\in T} p(t|d)\, p(w|t) = \sum_{t\in T} \phi_{wt}\theta_{td}. \qquad (1)$$

Learning the model parameters $\Phi = (\phi_{wt})$ and $\Theta = (\theta_{td})$ from the data (n_{dw}) is a problem of stochastic matrix factorization. This problem is ill-posed, since the set of its solutions is generally infinite. In the *additive regularization* (ARTM) framework, the appropriate solution is found from the regularized log-likelihood maximization under normalization constrains [26]:

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

$$\sum_{w\in W_m} \phi_{wt} = 1;\ \phi_{wt} \geq 0; \quad \sum_{t\in T} \theta_{td} = 1;\ \theta_{td} \geq 0;$$

where R_i are regularization criteria, τ_i are regularization coefficients, and τ_m are modality weights. The regularized variant of the EM-algorithm can be used to solve this optimization problem for any differentiable regularizers [25], [26]. Regularization coefficients τ_m and τ_i can be selected by coordinate-wise grid search.

In our experiments we use the combination of three regularizers that are known to improve both the interpretability of topics [25] and the search quality in terms of precision and recall [28].

1. *Decorrelation of term distributions in topics* first introduced in [23] makes the topics more different and interpretable by minimizing the sum of covariances between ϕ_t vectors. Besides, it helps to concentrate stop-words and common words in separate topics.

2. *Sparsing topic distributions in documents* enforces vectors θ_d be as far as possible from the uniform distribution thereby increasing the number of zero probabilities up to 90% and higher. This makes the document-by-document search more efficient.

3. *Smoothing term distributions in topics* compensates the excessive sparsing of ϕ_t vectors produced by decorrelation.

III. Hierarchical topic-based exploratory search

Exploratory search using conventional search engines requires a lot of time for multiple refinements of short queries. However, in many usage scenarios the search intent can be formulated beforehand in a form of a long text query. Looking at this text, the user makes many short queries to find relevant information more. Even when the user iteratively refines his search intent, it can be assumed that there were some initial texts that motivated his search activity. Thus, we aim to change the iterative nature of exploratory search and make it a quick one-step procedure. To do this, we use the document-by-document topic-based search. Having a long text query q the system learns its topic vector $p(t|q)$ in the same way as it was done for the documents in the collection. Next, the system ranks document vectors $p(t|d)$ by their similarity to the query and presents top k results to the user. The problem is to elaborate the topic-based search that proceeds in one step and gives better results than human experts might achieve with multiple queries. For the flat topic models, this has been demonstrated previously in [28]. In this work we are focused on hierarchical topic models with their ability to gradually narrow the scope of the search.

A hierarchical topic model divides topics into subtopics recursively [30]. We use a top-down level-by-level strategy proposed in [5] within the ARTM framework. Each level of the hierarchy is represented by a flat topic model. For each child level we find topic parents from the previous level using interlevel regularization. The regularizer claims parent topics to be well approximated by probabilistic mixtures of children's subtopics:

$$R(\Phi,\Psi) = \sum_{t\in T} \sum_{w\in W} n_{wt} \ln \sum_{s\in S} \phi_{ws}\psi_{st},$$

where conditional probabilities $\psi_{st} = p(s|t)$ link subtopics s with parent topics t. The maximization of $R(\Phi,\Psi)$ can be considered as topic modeling of $|T|$ pseudo-documents corresponding to the parent topics. Each pseudo-document contains all terms with counters $n_{wt} \propto \phi_{wt}$, which are known from the last iteration of the EM-algorithm for the parent level. This means that instead of implementing a regularizer for a topical hierarchy we can simply add the parent topic pseudo-documents into the collection before building the next level.

In the hierarchical topic-based search both query and document are represented by a sequence of topic vectors, one vector per the level. We compare query and document topic vectors level-by-level starting from the top-level vectors of

lower dimension and proceeding to the child-level vectors of higher dimensions. This helps to discard irrelevant documents gradually specifying the query from general to specific topics. This cascade-style search emulates the humans' natural strategy of information seeking. The elimination of irrelevant documents at top levels increases the precision and speeds up the search process.

IV. The evaluation of topic-based exploratory search

To evaluate exploratory search quality, we introduce new evaluation technique based on two-stage human assessments of relevance. First, we prepared a set of thematically focused long text queries gathering text fragments outside the collection. Each query should be short enough for an assessor to understand its meaning quickly, but it should also be sufficiently complete, so as to minimize discrepancies in its interpretation by different assessors. On average, a query should consist of roughly one A4 page of text.

At the first stage, assessor is asked to find within a given collection as many documents relevant to the query as possible. Assessor may use any search tools available, e. g. built-in search line, hyperlinks, tags or categories, conventional search system like Google, Bing, Yandex etc. The time taken to process a query is recorded.

At the second stage, assessor marks each document retrieved by the topic-based search for the same query as relevant or irrelevant.

Each query is processed by three assessors to reduce the variance of the result.

For each query we measure two quality metrics. Precision@k is the fraction of relevant documents among the first k documents found. Recall@k is the fraction of relevant documents found out of all the relevant documents. In order to evaluate the topic search quality we take the average precision and recall over all queries.

The calculation of Recall requires to know the set of all relevant documents for each query. We are approximating this set from below by joining the documents that were found by all assessors during both stages. At the second stage, we accept the document as relevant if the majority of assessors voted for it.

V. Experiments

A. Datasets

The experiments were based on two tech news collections: `TechCrunch.com` in English and `Habrahabr.ru` in Russian. Text pre-processing included deleting punctuation, bringing letters to the lower case, stemming for English texts (NLTK), and lemmatizing for Russian texts (pymorphy2).

The TechCrunch collection consists of 759324 articles. Articles contain terms of four modalities: 11523 word unigrams, 1.2 mln. bigrams (the tail of rare bigrams was deleted), 605 authors and 184 categories.

The Habrahabr collection consists of 175143 articles. Articles contain terms of six modalities: 10552 word unigrams, 742000 word bigrams, 524 authors, 10000 commentators (authors of comments to the articles), 2546 tags, 123 hubs (categories). We exclude 5% of the most frequent words in the collection.

B. Topic search vs. assessors

We applied the evaluation method described above to the Habrahabr and TechCrunch collections. For each collection we composed 100 queries by copying text fragments taken from external sources such as stackoverflow.com, ixbt.com, and other IT-oriented blogs. The length of the query ranges from 93 to 455 words with an average of 262 words for Habrahabr and from 75 to 392 words with an average of 195 words for TechCrunch. Each query was processed by three assessors.

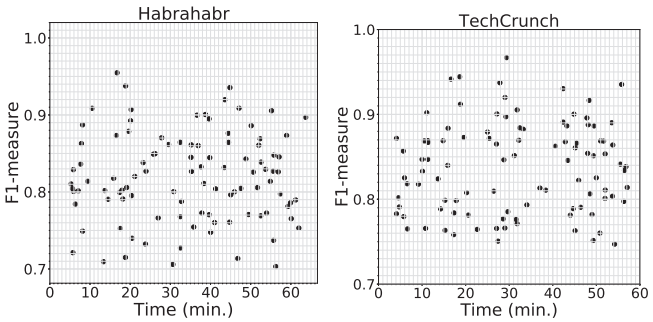

Fig. 1. The time (min.) spent by assessors to process each query

From the charts in Fig. 1 it can be seen that there is no obvious dependence between the time spent by an assessor and the quality of the search. On average it took assessors about 30 minutes to process a single query.

Topic-based search is trained in a fully unsupervised manner. This means that we use assessors' estimates only at validation stage, not for training the model, unlike conventional learning-to-rank models for information retrieval. That's why a so small number of queries is sufficient for our purposes without falling of the generalizing ability.

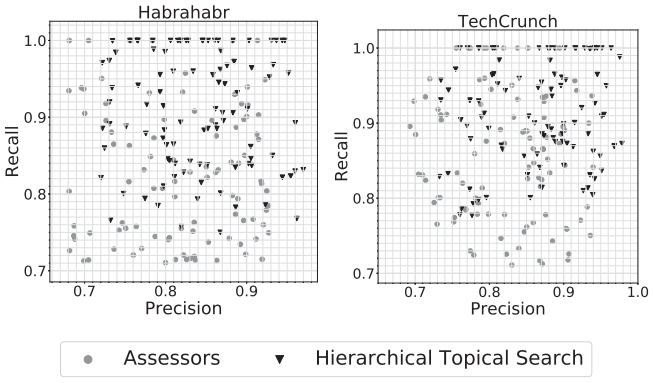

Fig. 2. The quality of assessors' and hierarchical topic-based search

The experiment results are presented in Fig. 2. The points on the plot correspond to queries. We compare precision and recall of the search performed by the assessors with the topic-based search for the best of our models (hierarchical ARTM with 3 levels, for more detail please refer to VI). On average,

precision for hierarchical topic-based search is 7% higher while recall is 10% higher than the same metric for manual human search. The same comparison between flat topic models and assessors' search shows just comparable precision and 5% higher recall.

The highest recall we got for the topic-based search is 1.0 for 26 queries out of 100 for Habrahabr and 29 queries out of 100 for TechCrunch. This means that our search engine is able to find documents that were missed out even by human annotators. Moreover, it took assessors from 15 to 65 minutes to process a single query while topic-based search gives answer in less than 1 sec. Thus, topic-based exploratory search obtains higher precision and recall and performs significantly faster than human assessors.

The difference in precision and recall between assessors search and topic-based search was tested to be statistically significant. We used Mann-Whitney test, which determines whether two samples stem from different distributions. As samples we take the values of precision/recall for all queries taken from the assessors search and the topic-based search. For all the tests p-value is less than 0.01.

C. Topic search vs. baselines

To prove the competitiveness of topic-based search we have compared it with several baselines. The text preprocessing steps for topic-based search and all the baselines remain the same as well as exploratory search evaluation technique covered in section IV. All the results for ARTM-based models and baselines are shown in 3.

The Mann–Whitney test confirmed that the differences between baselines and ARTM-based (both flat and hierarchical models) search are significant. P-values were less than 0.02 for each experiment with different combinations of metrics (Precision@k, Recall@k, $k \in \{5, 10, 15, 20\}$).

a) ***TF-IDF and BM-25****:* First, we use a simple but strong baseline based on TF-IDF. We transform a collection of raw documents and exploratory search queries to a matrix of TF-IDF features using simple TF-IDF vectorizer from skLearn library. Then we measure the similarity between TF-IDF representatons of query and documents the same way as we did it for topic-based approach. To make comparison with topic-based search fair we take into account not only words, but also bigramms and meta-information (tags and categories). We use the same n-gramm extractor TopMine [7] for all the baselines and topic search.

TF-IDF similarity search is a strong competitor for the topic-based search because it uses all the information about term frequencies whilst the topic-based search uses their approximate representation from the matrix factorization. Topic-based search gives better results in terms of precision and recall than the TF-IDF search. This fact confirms that our topic model is well-designed and gives better semantic representations of documents and queries. Another advantage of the topic-based search is the low-dimensional sparse topical representation, which enables to develop fast and cheap search services.

Also we used ranking function Okapi BM25 as a baseline. It performs very similar to TF-IDF baseline resulting in slightly higher precision and recall.

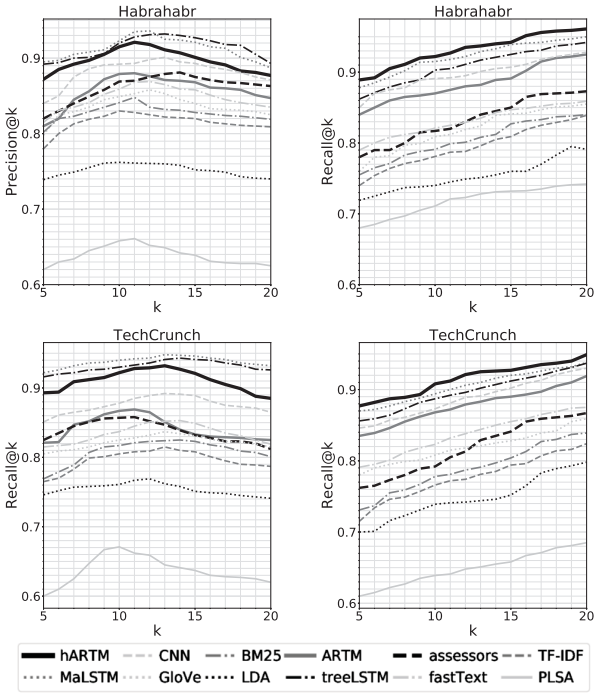

Fig. 3. Comparison between search performed by assessors, ARTM-based search and baselines

b) ***CNN-based approach****:* In [10] a CNN-based approach for measuring sentence similarity is proposed. Each sentence is modeled with a convolutional neural network that extracts features at multiple levels of granularity. Then the representations are compared using L2 and cosine metric (as well as in our approach). We reproduced the results from the paper for every sentence from our datasets (both for queries and documents) and then aggregate per-sentence representations to get vectors for the whole texts. Hierarchical ARTM beat up the CNN-based approach.

c) ***Word Embeddings****:* Neural word embedding approaches are able to capture semantics. Recently word embeddings have shown promising results in improving retrieval performance ([19]). These facts make word embeddings a good baseline to compare our work with.

First, we use pretrained vectors: GloVe.840B.300d for English texts and RusVectors (skipgram) trained on Russian Wikipedia for Russian texts. Then we applied fastText vectors because it is a good choice for languages with rich morphology especially Russian. Although word embeddings show comparable with manual human search quality, hierarchical ARTM outperform both GloVe and fastText in terms of precision and recall.

d) ***Tree LSTM****:* We apply Tree LSTM [22] for our queries and documents to compare it with ARTM-based search.

Our hierarchical searching algorithm has a comparable quality but computational expenses for building inverted topic index are much smaller than the ones for training Tree LSTM with 200000 weights ([22]).

e) ***Siamese adaptation of LSTM****:* Siamese adaptation of LSTM (MaLSTM) for assessing semantic similarity between sentences [16] supplement word embedding vectors with

synonymic information to the LSTMs which helps to uncover the underlying meaning expressed in a sentence. Although this technique is used for measuring sentences similarity, we have expanded its field of applicability to measuring distances between small texts (queries and documents) and used it as a baseline.

Precision of our hierarchical topic search is comparable with the one of MaLSTM. However, recall is a little higher for our approach.

What's interesting, in [16] Manhattan metric was used, while in our experiments cosine similarity gives slightly better results.

*f) **Other topic models**:* Also we compare the ARTM-based search over PLSA and LDA models as a baseline. They both perform worse than ARTM-based search. The text preprocessing steps for ARTM-based and PLSA/LDA models were the same.

VI. Improving the model without additional assessments

A. Topic models tuning techniques

Sets of relevant documents found by assessors for every query allow us to evaluate precision and recall of the topic search for new topic models without any additional assessments. It becomes possible to compare different topic models using search quality criteria.

There are several directions that can help to improve the model. Next we will show the results for hierarchical models only due to their better performance. The process of tuning parameters for flat models is identical. First of all, it is necessary to find the best similarity measure between query and documents from the collection in terms of search quality. We have tested five options: cosine similarity, euclidean distance, Manhattan distance, Hellinger distance, Kullback-Leibler divergence. For each of them we measured the precision@k and recall@k metrics. According to I, for both Habrahabr and TechCrunch tech news collections cosine similarity gives the best result.

The next challenge is to find an optimal number of topics for the model. Let's start with tuning the number of topics for unilevel model II. For Habrahabr collective blog articles we trained 7 models with $T \in [100, 150, 200, 250, 300, 400, 500]$ and found out that $T = 200$ gives the best search quality. For TechCrunch we did the same with $T \in [350, 400, 450, 475, 500]$.

Before tuning the number of topics for hierarchical model we need to determine an optimal number of levels. Models with more or equal then 4 levels have pure interpretation and also give very low quality ($pr@k < 0.72$, $r@k < 0.65$, $k \in [5, 10, 15, 20]$). This makes us choose between 2-level and 3-level models with different number of topics at each level. To find the best model we need to evaluate the quality of the overall model, not every level in alienation. In IV and VI we present several models with different number of topics on each level. Here we show just the best cases. However, our grid search for topic number included 75 parameter combinations.

Altering number of topics at the third level having the best combination for the first ones doesn't improve the quality dramatically. That is why we show third-level grid search results only for the best models. However, not having the third layer slightly decreases the quality. In tables IV and VI we show results for 3-level hierarchical models.

Next we compare multimodal topic models with different combinations of modalities. For first collection (Habrahabr) we were trying different sets of modalities out of five: terms (including words and collocations), comments, authors, tags and hubs (categories). In this experiment the number of topics was fixed and equal to $|T| = 200$ (the best one chosen in IV). For TechCrunch collection the set of modalities included terms (words and collocations), authors and categories. The number of topics was equal to 475. Table VII shows that using all modalities together improves precision and recall of the search significantly. Tag and term modalities contribute the most here. Models with only one modality show the worst results. In this experiment all models are hierarchical and have a fixed number of topics at each level.

B. Importance of regularizers

The goal of the next experiments is to prove that each regularizer is important and significantly improves the search quality. Table VIII shows that the decorrelation regularizer contributes the most to the search quality, but all other regularizers are also necessary. Model with no regularization gives much worse result than LDA and TF-IDF baselines.

We subsequently add regularizers to the model following empirical recommendations from [25]: decorrelation goes first, then smoothing and sparsing. Also we introduced interlevel connections regularizer for hierarchical models. For more information about regularizer trajectories please refer to [28].

C. Regularization trajectories

We use the combination of three regularizers: decorrelation of term distributions in topics, sparsing topic distributions in documents and smoothing term distributions in topics (see section II for more details).

It was shown [25] that the way we introduce regularizers significantly affects the model. Also choosing the best regularization coefficients requires huge and resource intensive grid-search which increases while introducing even one more regularizer to the model. That leads us to simple yet efficient technique of finding optimal regularization trajectory.

We add regularizers to the model one by one in the indicated order: decorrelation goes first, than Φ smoothing and Θ sparsing. As we add each regularizer its coefficient is chosen from a grid of values using several topic model quality criteria (perplexity, Φ and Θ sparsity). For each value of a regularization coefficient we do 8 iterations of the EM-algorithm. Among these we choose the one that yields an improvement in at least one of the criteria without a significant impairment by the others.

VII. Conclusion

The main goal of exploratory search is intensification and automation of acquisition and systematization of knowledge.

TABLE I. TOPIC SEARCH WITH DIFFERENT SIMILARITY MEASURES: 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

TABLE II. FLAT TOPIC MODELS WITH DIFFERENT NUMBER OF TOPICS

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

TABLE III. 3-LEVEL HIERARCHICAL TOPIC-BASED SEARCH WITH DIFFERENT NUMBER OF TOPICS (HABRAHABR)

$\|T_1\|$	**20**		**25**							**30**	
$\|T_2\|$	150	200	**250**		**275**			**300**		400	450
$\|T_3\|$	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

TABLE IV. 2-LEVEL HIERARCHICAL TOPIC-BASED SEARCH WITH DIFFERENT NUMBER OF TOPICS (HABRAHABR)

$\|T_1\|$	**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

Topic modeling is regarded as one of the key technologies for exploratory search. In this paper we investigate exploratory topic search with long text queries and test the proposed method on the Habrahabr and TechCrunch text collections of tech news articles.

We have studied both flat and hierarchical topic models applied to exploratory search and showed that hierarchical models as well as cascade interlevel search allows to get impressive results including average recall being higher than 95%. We descried that iterative level-by-level search emulates exploratory search nature with its gradual query rephrasing in order to clarify search intent. All the topic models were built using open-source library BigARTM, which allows to optimize several quality criteria simultaneously and find low-dimensional topic representations.

To measure the quality of search we proposed an evaluation technique and developed a special collection of queries for exploratory search, which were processed both by assessors and topic search engine. The relevance of the found documents was again evaluated by the assessors. This method has a unique property: once you've done the markup of the search results by assessors, you may repeatedly evaluate other topic models and topic search engines based on them.

TABLE V. 3-LEVEL HIERARCHICAL TOPIC-BASED SEARCH WITH DIFFERENT NUMBER OF TOPICS (TECHCRUNCH)

$\|T_1\|$	**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

TABLE VI. 2-LEVEL HIERARCHICAL TOPIC-BASED SEARCH WITH DIFFERENT NUMBER OF TOPICS (TECHCRUNCH)

$\|T_1\|$	**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

TABLE VII. HIERARCHICAL TOPIC-BASED SEARCH USING DIFFERENT MODALITIES
HABRAHABR: ASSESSORS, WORDS, BIGRAMS, COMMENTS, TAGS, HUBS, AUTHORS
TECHCRUNCH: ASSESSORS, WORDS, BIGRAMS, AUTHORS, CATEGORIES

	Habrahabr						**TechCrunch**					
	As	W	C	WB	WBTH	**All**	As	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**

TABLE VIII. TOPIC-BASED SEARCH WITH DIFFERENT SET OF REGULARIZERS:
DECORRELATION, Θ-SPARSING, Φ-SMOOTHING, INTERLEVEL CONNECTIONS SPARSING

	Habrahabr					**TechCrunch**				
	no reg	D	DΘ	D$\Theta\Phi$	D$\Theta\Phi$I	no reg	D	DΘ	D$\Theta\Phi$	D$\Theta\Phi$I
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**

The experiments have shown the advantages of hierarchical topic-based search over manual human search in terms of precision (7%) and recall (10%). Also topic-based search is able to provide a result much faster than assessors. To prove the competitiveness of our approach against other methods we compared our hierarchical topic-based search with several baselines including TF-IDF, word embeddings (pretrained GloVe and fastText), CNN-based methods, tree LSTM and siamese adaptation of LSTM (MaLSTM). Topic-based search outperformed all the baselines in terms of recall. Also it showed comparable quality in terms of precision for tree LSTM and MaLSTM and precision being higher by more than 4% for the rest of the baselines.

Finally, we provide technical details regarding topic models training process to make reproducing our results easy. Tuning number of levels and topics per level as well as trying different similarity measures gives additional insights about choosing the best model. Moreover, tuning model by criteria of precision and recall of the search showed that including meta-

information like tags and categories significantly improves the search quality while meta-information about the authors and comments gives a negligible increase in quality.

REFERENCES

[1] D. Andrzejewski and D. Buttler. "Latent Topic Feedback for Information Retrieval", *in Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '11*, 2011, San Diego, California, USA, pp. 600–608.

[2] M. Apishev, S. Koltcov, O. Koltsova, S. Nikolenko, and K. Vorontsov. "Additive regularization for topic modeling in sociological studies of user-generated text content", *MICAI 2016, 15th Mexican International Conference on Artificial Intelligence*, vol.10061, 2016, pp. 166-181. Springer, Lecture Notes in Artificial Intelligence.

[3] D.M. Blei. "Probabilistic topic models", *Communications of the ACM*, 55(4), 2012, pp. 77-84.

[4] D.M. Blei, A.Y. Ng, and M.I. Jordan. "Latent Dirichlet allocation", *Journal of Machine Learning Research*, vol.3, 2003, pp. 993-1022.

[5] N.A. Chirkova and K.V. Vorontsov. "Additive regularization for hierarchical multimodal topic modeling", *Journal Machine Learning and Data Analysis*, vol.2(2), 2016, pp. 187-200.

[6] Ali Daud, Juanzi Li, Lizhu Zhou, and Faqir Muhammad. "Knowledge discovery through directed probabilistic topic models: a survey", *Frontiers of Computer Science in China*, vol.4(2), 2010, pp. 280-301.

[7] Ahmed El-Kishky, Yanglei Song, Chi Wang, Clare R. Voss, and Jiawei Han. "Scalable 883 topical phrase mining from text corpora", *Proc. VLDB Endowment*, vol.8(3), 2014, pp. 305-316.

[8] O.Frei and M.Apishev. "Parallel non-blocking deterministic algorithm for online topic modeling", *in AIST2016, Analysis of Images, Social networks and Texts*, vol.661, 2016, pp. 132-144. Springer International Publishing Switzerland, Communications in Computer and Information Science (CCIS).

[9] Ch. E. Grant, C. P. George, V. Kanjilal, S. Nirkhiwale, J. N. Wilson and D. Zhe Wang. "A Topic-Based Search, Visualization, and Exploration System". *in FLAIRS Conference*, AAAI Press, 2015, pp. 43–48.

[10] Hua He, Kevin Gimpel, and Jimmy Lin. "Multi-perspective sentence similarity modeling with convolutional neural networks", in Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 2015, pp. 1576-1586. Association for Computational Linguistics.

[11] Th. Hofmann. "Probabilistic latent semantic indexing", *in Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, 1999, pp. 50-57, New York, NY, USA. ACM.

[12] D. Kochedykov, M. Apishev, L. Golitsyn, and K. Vorontsov. "Fast and modular regularized topic modelling", *in Proceeding Of The 21St Conference Of FRUCT (Finnish-Russian University Cooperation in Telecommunications) Association, the seminar on Intelligence, Social Media and Web (ISMW)*, 2017, Helsinki, Finland, November 6-10, pp. 182-193. IEEE.

[13] W. Kraaij and W. Post. "Task based evaluation of exploratory search systems", *in Proceedings SIGIR 2006 workshop on Evaluating Exploratory Search Systems (EESS)*, 2006, pp. 24-27. ACM.

[14] C.D. Manning, Pr. Raghavan, and H.Schutze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.

[15] G. Marchionini. "Exploratory search: From finding to understanding", *Commun. ACM*, 2006, vol.49(4), pp. 41-46.

[16] J. Mueller and A. Thyagara. "Siamese recurrent architectures for learning sentence similarity", *in Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, AAAI16*, 2016, pp. 2786-2792. AAAI Press.

[17] M. Potthast, M. Hagen, M. Volske, and B. Stein. "Exploratory search missions for TREC topics", *in EuroHCIR, volume 1033 of CEUR Workshop Proceedings*, 2013, pp. 7-10.

[18] S. Ronnqvist. "Exploratory Topic Modeling with Distributional Semantics", *Advances in Intelligent Data Analysis XIV: 14th International Symposium, IDA 2015*, Saint Etienne, France, 2015, pp. 241–252.

[19] D. Roy, D.s Ganguly, S. Bhatia, S. Bedathur, and M. Mitra. "Using word embeddings for information retrieval: How collection and term normalization choices affect performance", *in Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 18*, 2018, pp. 1835-1838, New York, NY, USA. ACM.

[20] M. Scherer, T. von Landesberger T. Schreck. "Topic Modeling for Search and Exploration in Multivariate Research Data Repositories", *in Research and Advanced Technology for Digital Libraries: International Conference on Theory and Practice of Digital Libraries*, TPDL 2013, Valletta, Malta, 2013, pp. 370–373.

[21] Ch. Shah, Ch. Hendahewa, and R. Gonzalez-Ibanez. "Rain or shine? Forecasting search process performance in exploratory search tasks", *Journal of the Association for Information Science and Technology*, 2016, 67(7), pp. 1607- 1623.

[22] K.Sh. Tai, R. Socher, and C.D. Manning. "Improved semantic representations from tree-structured long short-term memory networks", *in Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2015, pp. 1556-1566. Association for Computational Linguistics.

[23] Y. Tan and Zh. Ou. "Topic-weak-correlated latent Dirichlet allocation", *in 7th International Symposium Chinese Spoken Language Processing (ISCSLP)*, 2010, pp. 224-228.

[24] E.E. Veas and C. di Sciascio. "Interactive Topic Analysis with Visual Analytics and Recommender Systems", *in 2nd Workshop on Cognitive Computing and Applications for Augmented Human Intelligence, CCAAHI2015, International Joint Conference on Artificial Intelligence, IJCAI*, Buenos Aires, Argentina, 2015.

[25] K.V. Vorontsov and A.A. Potapenko. "Additive regularization of topic models", *Machine Learning, Special Issue on Data Analysis and Intelligent Optimization with Applications*, 2015, 101(1), pp. 303-323.

[26] K.V. Vorontsov, O.Frei, M. Apishev, P. Romov, M. Suvorova, and A. Yanina. "Non-bayesian additive regularization for multimodal topic modeling of large collections", *in Proceedings of the 2015 Workshop on Topic Models: Post-Processing and Applications*, 2015, pp. 29-37, New York, NY, USA. ACM.

[27] R.W. White and R.A. Roth. "Exploratory Search: Beyond the Query-Response Paradigm", *Synthesis Lectures on Information Concepts, Retrieval, and Services*, 2009, Morgan and Claypool Publishers.

[28] A. Yanina, L. Golitsyn, and K. Vorontsov. "Multi-objective topic modeling for exploratory search in tech news", *in Communications in Computer and Information Science*, vol.789. AINL-6: Artificial Intelligence and Natural Language Conference, St. Petersburg, Russia, 2017, pp. 181-193. Springer International Publishing, Cham.

[29] X. Yi and J. Allan. "A Comparative Study of Utilizing Topic Models for Information Retrieval", *in Advances in Information Retrieval, Lecture Notes in Computer Science*, 2009, vol.5478, pp. 29–41.

[30] E. Zavitsanos, G. Paliouras, and G.A. Vouros. "Non-parametric estimation of topic hierarchies from texts with hierarchical Dirichlet processes", *Journal of Machine Learning Research*, 2011, vol.12, pp. 2749-2775.