QUALITY EVALUATION AND IMPROVEMENT FOR HIERARCHICAL TOPIC MODELING

Belyy A. V.[1,2] (anton.belyy@gmail.com), Seleznova M. S.[3] (maria.selezniova@phystech.edu), Sholokhov A. K.[3] (ak.sholokhov@gmail.com), Vorontsov K. V.[3] (vokov@forecsys.ru)

[1] ITMO University, Saint Petersburg, Russia; [2] B Tochka Bank QIWI Bank (JSC), Yekaterinburg, Russia; [3] Moscow Institute of Physics and Technology (State University), Moscow, Russia

Generic topics of large-scale document collections can often be divided into more specific subtopics. Topic hierarchies provide a model for such topic relation structure. These models can be especially useful for exploratory search systems. Various approaches to building hierarchical topic models have been proposed so far. However, there is no agreement on a standard approach, largely due to the lack of quality metrics to compare existing models. To bridge this gap we propose automated evaluation metrics which measure the quality of topic-subtopic relations (edges) of a topic hierarchy. We compare automated evaluations with human assessment to validate the proposed metrics. Finally, we show how the proposed metrics can be used to control and to improve the quality of existing hierarchical models.

Key words: topic modeling; topic hierarchies; quality metrics; coherence; word embeddings.

ОЦЕНКА И УЛУЧШЕНИЕ КАЧЕСТВА ИЕРАРХИЧЕСКИХ ТЕМАТИЧЕСКИХ МОДЕЛЕЙ

Белый А. В.[1,2] (anton.belyy@gmail.com), Селезнева М. С.[3] (maria.selezniova@phystech.edu), Шолохов А. К.[3] (ak.sholokhov@gmail.com), Воронцов К. В.[3] (vokov@forecsys.ru)

[1] Университет ИТМО, Санкт-Петербург, Россия; [2] Ф Точка Банк КИВИ Банк (АО), Екатеринбург, Россия; [3] Московский физико-технический институт (государственный университет), Москва, Россия

1 Introduction

Topic modeling is a branch of unsupervised machine learning widely used to summarize large unlabeled text corpora. A probabilistic topic model extracts latent probabilities of words appearing in each topic and topics appearing in each document, uncovering vectors of probability distributions that represent documents.

For the purposes of creating a representation of a text collection that helps users to navigate through the collection smoothly, topics can be arranged into a hierarchy. Generic topics of each parent level are thus divided into more specific subtopics of its child level. Such representation allows users to constrict the set of documents they are interested in gradually going down the topic hierarchy.

Various approaches to topic hierarchy learning have been proposed in recent years, such as LDA [1], hPAM [2] and hARTM [3]. However, there is still no agreement on the common approach. The main problem resides in difficulties of topic hierarchies comparison. Since there is no common topic hierarchy quality metrics, it is currently impossible to compare different approaches rigorously.

A quality metric for hierarchical topic models should measure both interpretability of topics on each hierarchy level and quality of pairs of topics that are linked with parent-child relations in the hierarchy. There are common ways to measure topics quality widely used in the field, such as topic coherence [4]. Also, various topic quality metrics based on word embeddings have been proposed recently [5, 6]. However, to the best of our knowledge, "parent-child" relations quality has not been explored so far. In this paper, we propose metrics for quality of the hierarchy edges which represent such relations.

We use BigARTM – an open source library for topic modeling of large collections – in our experiments.

2 Hierarchical Topic Models

Let D denote a document collection. A vocabulary W is a set of tokens (e.g. words, tags, links, etc.) that appear in the collection. We assume that the collection contains topics from a finite set T. Then, each document d ∈ D can be described with its probability distribution p(t|d) over the topics t ∈ T (i.e. p(t|d) is a vector of probabilities for each topic to appear in the document d). On the other hand, each topic t ∈ T is described with its probability distribution p(w|t) over the tokens w ∈ W.

Given a collection D, we can extract estimators of a probability distribution p(w|d) of its tokens over its documents as $\frac{n_{dw}}{n_d}$, where n_{dw} is a number of times the token w appears in the document d, $n_d = \sum_{w \in W} n_{dw}$ is a number of words in d. However, we cannot directly estimate p(w|t) or p(t|d) as t is a latent (hidden) variable. As described, for example, in [12], extracting those distributions can be formulated as a matrix factorization problem

$$F = \Phi\Theta,$$

where $F = [p(w|d)]_{W \times D}$ is the given matrix of p(w|d) estimators, $\Phi = [p(w|t)]_{W \times T}$ and $\Theta = [p(t|d)]_{T \times D}$ are the matrices of model parameters that we are aimed to find. As shown in [12], the problem can be solved through EM-algorithm application.

A hierarchical topic model (HTM) comprises several flat (described above) topic models that form hierarchy levels. Each (l+1)-th level has more topics than the l-th one for the topics to get more specific down the hierarchy. HTM also includes edges that represent “parent-child” relations between the topics of the neighboring levels. As shown in [3], the problem of building such a hierarchy level by level can be solved through adding a new matrix $\Psi^l = [p(t|a)]_{T \times A}$ that represents probabilities for topics t $\in T^{l+1}$ (a set of topics of the (l+1)-th level) to be subtopics of the previous level topics a $\in T^l$. That gives a matrix factorization problem

$$\Phi^{l+1} = \Phi^l \Psi^l$$

for each level [3].

3 Motivation

Fig. 1. Start screen of the exploratory search system Rysearch.

Since its inception, topic modeling has been successfully applied for visualizing and navigating through large scientific corpora [8, 9, 10]. flexibility for such visualizations by allowing a user to go down a hierarchy to more specific topics or go up to more general ones. HTMs are particularly promising as a technology for creating an exploratory search engine [11], which allows exploring an area of knowledge related to a user's query rather than looking for the exact query.

Our distant goal was to create such an engine. We have been working on a prototype, which currently indexes Russian popular science websites and blogs and aggregates their content into a hierarchical topic model. In our work, we have faced two major problems:

- **heterogeneity of sources**, which means that our sources usually differ in size and comprise different sets of topics,
- **absence of evaluation metrics for HTMs** for HTMs, which slows down model design as each HTM needs to be evaluated manually before it can be deployed into production environment.

In this paper we propose several metrics for automated model evaluation, thus tackling the second problem. We also share some insights on how these metrics can be used to improve quality of already built hierarchical model.

To facilitate comparison of metrics, we built two hierarchical models, **concat** and **heterogeneous**, which we use throughout the paper for explanations. The first one is intentionally worse in subjective quality than the latter: it has been trained on a simple concatenation of all the sources into a single one, disregarding the inequality of their sizes and structure. The latter model was built using a method we proposed that includes several stages:

1. Build a topic model of a 'base' collection -- a collection that allows to build an interpretable model that includes all the topics we want to aggregate -- to create an initial estimate for the hierarchy.
2. Rank documents of a collection we want to add to the hierarchy according to their similarity to the base collection. The most similar documents should be ranked first.
3. Add the documents that appeared to be on the top of the ranking list to the base collection in a quantity not exceeding 10% of the base collection size. Build a topic model of the extended collection using the matrix Φ_0^1 from the first stage (that contains estimates for the first level topics of the base collection) to initialize a new matrix Φ^1 of the first hierarchy level.
4. Repeat the third step until all the documents from new collection are added to the model. Each time the collection from the previous iteration of the method is referred to as a base collection, so its size increases and we can add more documents on each step.

In our experiments we used Postnauka.ru as a base collection and Habrahabr.ru as an added collection. Postnauka.ru contains all the major topics present in popular science content and its' articles have manually adjusted tags that allow building a model of high interpretability. On the other hand, Habrahabr.ru is focused on IT and contains a lot of irrelevant content such as news and advertisements. Also, Habrahabr.ru collection is much bigger than Postnauka.ru (see the section 5.1 for the detailed datasets description). To rank the documents we used a regressor that measured how similar a given document is to typical popular science articles. As our ranking method is collection-specific we mention some other ideas about ways to perform ranking in the discussion section.

4 Proposed metrics for hierarchies

Proposed in [4], the topic coherence is a classical measure of a topic quality as well as flat topic models' interpretability in general. In particular, one can estimate quality of a model as a whole by taking the average of topics' coherences. However, a hierarchical topic model consists not only of its levels, but also of relations between topics from the neighboring levels, whereas the average coherence of the model's levels takes these dependencies completely out of consideration. Hence, the average coherence fractionally depicts the quality of a hierarchical model. This section is aimed to bridge the gap by proposing several quality measures for the "parent - child" relations between topics in a hierarchical model.

Linguistic similarity based metrics. We extend the classical flat coherence from [4] to hierarchical coherence to capture either syntagmatic or paradigmatic relatedness of parent and child topics' top tokens [7]. Let us define $w_i^{(t)}$ as the i-th top token of some topic t. Then $v_i^{(t)}$ will be the vector

corresponding to this top token in some Vector Space Model (VSM). In our experiments we used the pre-trained VSM RusVectōrēs [13], which was trained on Russian National Corpus and Russian Wikipedia (600 million tokens, resulting in 392 000 unique word embeddings). $D(w_1, w_2)$ is a number of documents in some corpus (in our experiments we use Postnauka corpus to calculate cooccurrences, although in general it is more preferable to use big external corpora such as Wikipedia or Twitter) where words w_1 and w_2 have occurred together at least once. D(w) is a document frequency of word w calculated for the same corpus. Then we define our metrics as:

- EmbedSim: $\frac{1}{C}\sum_{i=1}^{n}\sum_{j=1}^{n} < v_i^{(a)},\ v_j^{(t)} > [w_i^{(a)} \neq w_j^{(t)}]$,
- CoocSim: $\frac{1}{C}\sum_{i=1}^{n}\sum_{j=1}^{n} ln\frac{D(w_i^{(a)}, w_j^{(t)}) + \varepsilon}{D(w_j^{(t)})}[w_i^{(a)} \neq w_j^{(t)}]$,

where C = $\sum_{i=1}^{n}\sum_{j=1}^{n}[w_i^{(a)} \neq w_j^{(t)}]$ is a number of word pairs excluding pairs of identical words. We denote the topic of the parent level as t and the topic of the child level as a (“ancestor”) here, n is the number of considered top tokens for each topic.

Probabilistic similarity based metrics. We can compare parent and child topics as probability distributions. Two standard similarity measures for distributions P and Q are Hellinger distance and Kullback-Leibler divergence. The first one is a bounded metric and can be interpreted as distance between two topics in some space. The second is an unbounded asymmetric measure and can be interpreted as "how much information will be lost if we substitute parent topic P with some child topic Q".

- HellingerSim: 1 - $\frac{1}{\sqrt{2}}\ ||\sqrt{p(w|t)} - \sqrt{p(w|a)}||_2$
- KLSim: $-D_{KL}(p(w|a)||p(w|t))$

To understand how these metrics work, let us consider an example. We are given 6 “parent-child” pairs of topics that were assessed by humans. Three of them are labeled as **’good’** (there is a semantic similarity between parent and child), other three are labeled as **’bad’** (little or no similarity). On the fig. 2 one can see these pairs on the right, along with their scores given by the EmbedSim metric. The higher the score, the more confident the metric is. On the left there is a distribution of all the edges from an assessment task described in the following section. Y-coordinates of points are assigned according to metric score, and colors are set by the assessment experts. One can see that there are much more pairs marked as ‘bad’ than the ones marked as ‘good’. As we expect the hierarchy to be sparse (each parent topic has only a few suitable subtopics), this observation corresponds to our expectations. It is also clear from the figure that ‘bad’ pairs have lower average metric score than the ‘good’ ones. It means that the EmbedSim metric scores ‘good’ pairs with higher values and, therefore, correlates with the assessors opinion.

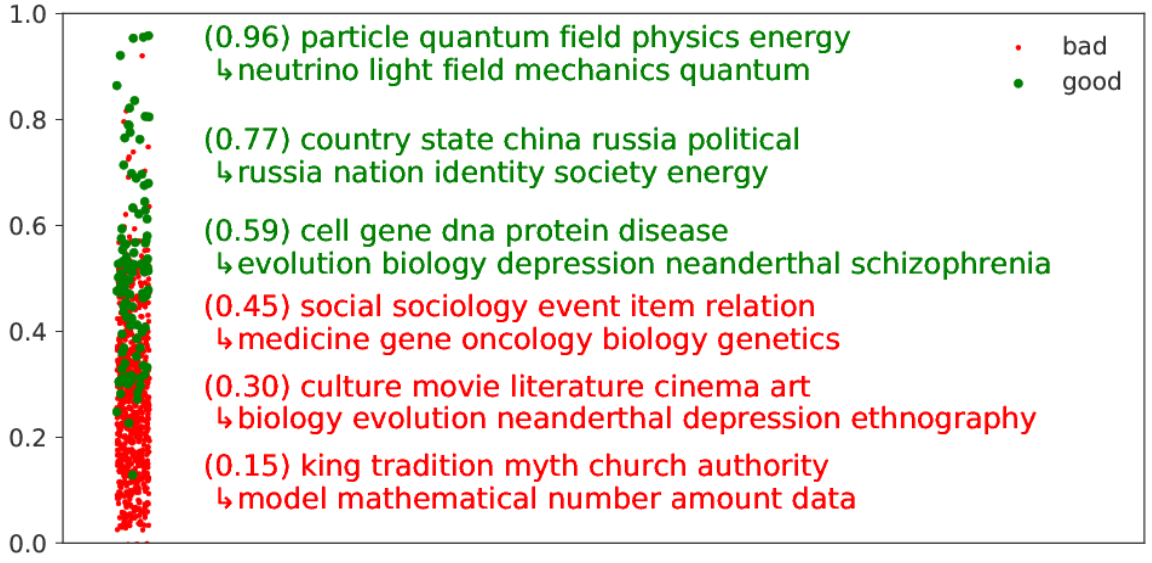

Fig. 2. Examples of topic-subtopic pairs from the assessment task (section 5) scored with the EmbedSim metric as hierarchy edges. Each topic or subtopic is represented by its 5 top tokens. The column on the left shows all the assessed pairs as dots on their EmbedSim score scale.

5 Expert opinions of edges quality

5.1 Datasets and models

To construct "parent-child" topic pairs for human annotation, we trained three two-level hierarchical topic models on three datasets:

- **Postnauka.ru**, a popular science website with edited articles on a wide spectrum of topics, focusing on humanities,
- **Habrahabr.ru** and **Geektimes.ru**, social blogging platforms specializing in Computer science, engineering and IT entrepreneurship,
- **Elementy.ru**, a popular science website with a particular focus on life sciences.

Dataset	Number of documents	Unique words	Unique tags	Parent topics	Child topics
Postnauka	2976	43196	1799	20	58
Habrahabr	81076	588400	77102	6	15
Elementy	2017	30352	-	9	25

Table 1. Datasets' descriptions. The collections consist of text documents. Dictionary sizes for each collection are listed in the "Unique words" column. Postnauka and Habrahabr collections are also manually tagged by their authors or editors (each article can have multiple tags), the numbers of tags are listed in the "Unique tags" column.

5.2 Task statement

The following question was asked for experts: "given two pairs of topics, T_1 and T_2, decide whether one is a subtopic of another". Possible answers were: "T_1 is a subtopic of T_2", "T_2 is a subtopic of T_1 " and "These topics are not related". Topic t was denoted by 10 top words from its probability distribution p(w|t).

After the experiment was finished, the first two answers were grouped to denote a single answer "These topics are somehow related" as it was often difficult for assessors to distinguish between a parent and a child given their top words.

5.3 Quality control

To ensure quality control, only those workers who completed training were allowed to enter the assessment task. Experts could have skipped some tasks if they were not sure, but those who were skipping tasks too often were banned from participating for a day.

5.4 Results

Overall, 68 trusted workers participated in our study, each contributed around 100 assessed topical pairs. Assessment of one pair of topics, given their 10 top words, took around 5 seconds for each

participants on average. Each topic pair was evaluated by at least five different experts, which gave us 6750 expert annotations for 1350 unique pairs (edges).

Our participants were mainly Russian and Ukrainian nationals, with age varying from 21 to 64 years.

Agreed assessors	Edge count	Edge percentage
3	374	27.7%
4	468	34.7%
5	508	37.6%

Table 2. Inter-assessor agreement. For each pair of topics, we calculate how many assessors made the same verdict (that the topics from the pair are related or that they are not). For 5 assessors per pair, there is always a majority decision, but it can be reached by either 3, 4, or 5 assessors. In the second and the third column we show the quantity and the percentage of the edges with the number of agreed assessors from the first column.

6 Comparison of metrics values and expert annotations

If many people think that there is a “topic – subtopic" relation between two particular topics in a model, a good metric should give a high score for such pair of topics. In this case we say that metric “approximates assessors opinion”. Moreover, we want that metric to keep an order on the model edges consistent with this statement: the more people agreed that the relation presents – the higher the metric score should be.

In order to prove that the proposed metric holds this constraint, consider the following classification problem. Let us call “the assessors’ judgment” the fact that 4 or 5 assessors agreed on the same verdict (that an edge exists or does not exist in a hierarchy). If it holds, then assessors’ judgment on this edge is equal to 1, and -1 otherwise. Let the edges of a hierarchical model be the objects: the positive and negative classes consist of the edges with a positive and a negative assessors’ opinion respectively. Let the classifier based on the metric be the following:

$$f(t_1, t_2) = \text{sign}(\rho(t_1, t_2) - w)$$

where t_1 and t_2 are the topics from parent and child level of the model respectively, ρ is one of the proposed metrics and w is a margin of the classifier. Having it written in this form, we can calculate ROC AUC for each classifier and estimate the quality of each metric: better approximators are expected to have better scores.

Metric	Score
EmbedSim	0.878
CoocSim	0.815
KLSim	0.790
HellingerSim	0.766

Table 3. ROC AUC scores for the proposed metrics.

The table 3 presents ROC AUC score for each classifier. One can see that the best classification quality was demonstrated by the classifier based on the Embeddings metric (AUC = 0.878). The other metrics demonstrated moderate yet acceptable consistency with the assessors’ opinion: AUC values

lied evenly above 0.75. For better understanding of this result one can see the fig. 3. For each graph the red line is a density distribution of the metric value for bad edges, and the green one is the same for good edges. The better some vertical line divides bad edges from good ones – the better the metric is. In further experiments we use the EmbedSim metric, as this metrics demonstrated the best consistency with assessors' judgment.

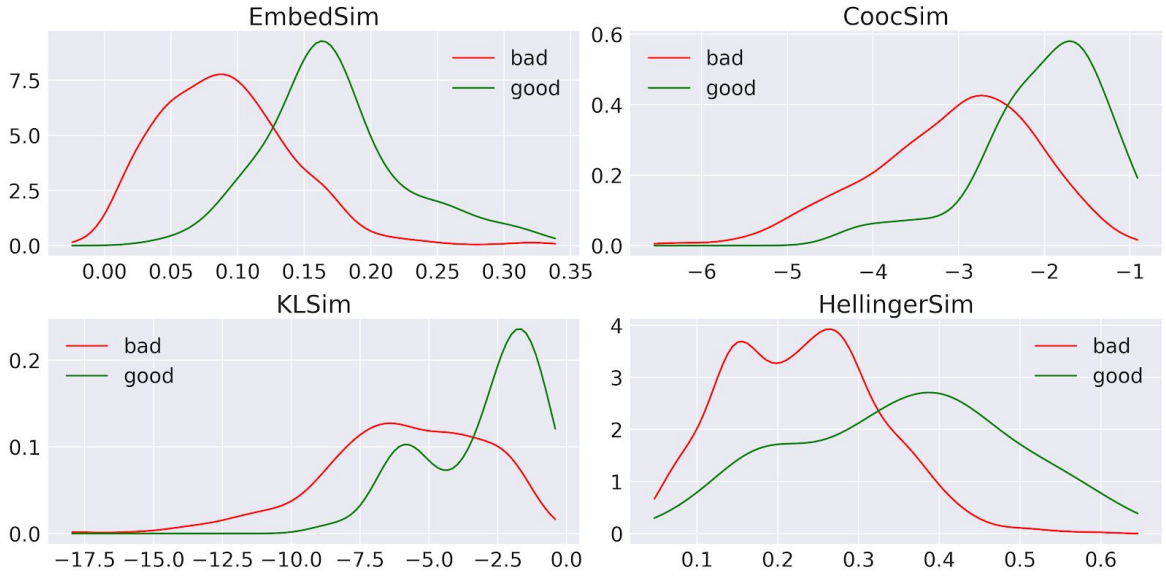

Fig. 3. Distribution of scores for 'bad' and 'good' topical edges.

As the EmbedSim metric gives the highest AUC score we use it in all the following experiments.

7 Quality of hierarchical models

The goal now is to combine the edges metric into some construction, which would be a representative quality measure for a hierarchy as a whole.

Normalization: Hereafter we work with a normalized matrix Ψ^{norm} as the following:

$$\Psi_{ta}^{norm} = \frac{\Psi_{ta} - min_t \Psi_{ta}}{max_t \Psi_{ta} - min_t \Psi_{ta}}$$

It allows to apply shared topic-agnostic threshold and to rank all values of Ψ matrix on the same scale.

7.1 Averaging quality

In the spirit of [4] where the average topics coherence was used as a model quality measure, let us consider the average edge quality as quality measure for our hierarchy. The particular hierarchy configuration depends on the chosen threshold for Ψ^{norm}, which determines what probability $p(t|a)$ is sufficient to include an edge connecting t and a into the hierarchy. Therefore different thresholds lead to different values of a quality measure. However, for bad models this value seems to be almost evenly lower than for good ones. The fig. 4 illustrates this effect: one can see that the heterogeneous model had a higher score than the less elaborated concat model no matter what threshold was set. Hence it was enough to set the same threshold for all models if one wants to compare them with our measure. However, this measure lacks the interpretability of its value (Y-coordinate of curves on the figure).

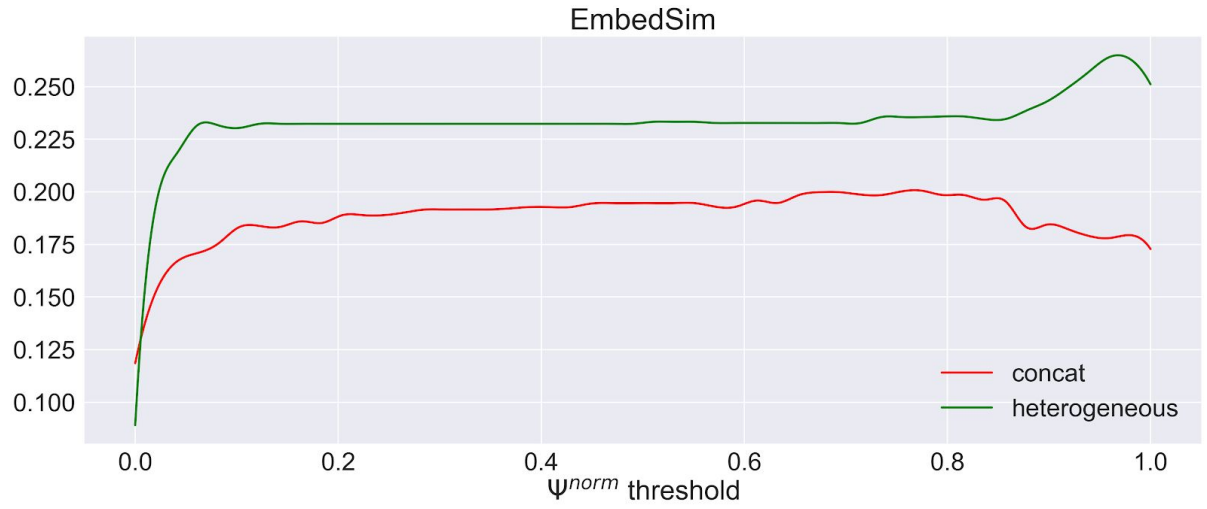

Fig. 4. Averaging quality metrics for EmbedSim.

7.2 Ranking quality

Another approach to form a quality measure with an interpretable value is to consider the process of establishing a hierarchy as a ranking process. Consider that we have built a model i.e. we have matrices Ψ^{norm}, Θ and Φ for each level. It would be natural to accept only the most meaningful edges according to a human's point of view. As our edge metrics turned out to be good approximators of the assessors' judgment, we can choose only k edges with the top scores of some fixed metrics. If our model is "good", then top-k scored edges (let us call them "the request") should match with top-k maximal elements of the Ψ^{norm} matrix (let us call them "the response"). The difference between the request and the response for each k was measured by common ranking metrics, such as :

- Average Precision (AP@k) -- described in [14].
- Inverse Defect Pairs (Inverse DP@k) -- the inverse value of the number of pairs that appear in the wrong order (i.e. are reversed) in the response.
- Normalized Discounted Cumulative Gain (nDCG@k) -- described in [14].

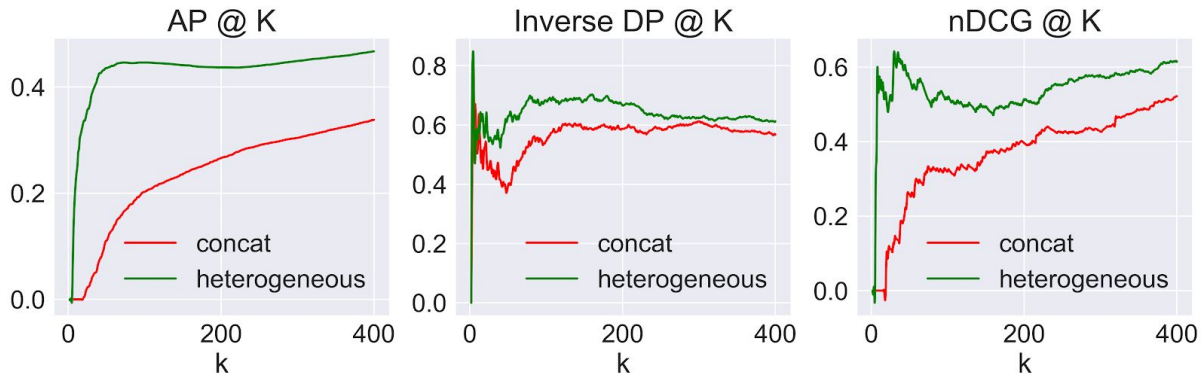

Fig. 5. The ranking quality metrics for the EmbedSim edge metric.
The considered models (concat and heterogeneous) are described in section 3.

The fig. 5 shows that in all cases the ranking quality scores are higher for better model. One may interpret this result as the following: if a model is "good", than its top-k edges should match the metric's

top-k edges precisely enough, no matter what k was set. According to the fiig. 4 it holds for all ranking metrics, but the biggest gap was given by the Average Precision. Hence, if one wants to compare quality of two different hierarchies, the advice may be the following:

➔ Take Embedding similarity (EmbedSim) as the edge metric and plot the Average Precision@k graph. The better model will be the one having better score(s) at the desirable value(s) of k.

There is also a notable advantage of ranking approach over the averaging approach: it allows to choose the optimal number of edges in the model, which we will discuss in the following section.

8 Applications

Using the proposed edge metrics we managed to improve the quality of models significantly in our own project mentioned in section 3. The following chapter briefly describes our experience and results.

8.1 Validating the model in an automated way

```
topic_1: file server application user data
    ↳ design internet server file electronics
    ↳ file browser query service web
topic_5: star universe galaxy hole black
    ↳ menu map download mass black
    ↳ astrophysics space gravitation telescope sun
    ↳ space astrophysics sun galaxy gravitation
    ↳ space galaxy telescope astrophysics sun
topic_6: cell gene dna organism protein
    ↳ linux species test result arise
    ↳ evolution bioinformatics paleontology cell zoology
    ↳ dna evolution cell genomics rna
    ↳ evolution ethology cell ecosystem adaptation
    ↳ buffer information open newsletter meditation
    ↳ cell mutation protein dna os
topic_8: brain child neuron memory emotion
    ↳ brain psychology memory neuron thinking
topic_11: country state political authority politics
    ↳ politics state ussr political science authority
```

Fig. 6. Examples of "good" (green), "bad" (red) and "moderate" (pink) edges according to the EmbedSim metric. Each topic or subtopic is represented by its 5 top-tokens.

Let us suppose that we have a service that continuously aggregates information from various sources into one heterogeneous HTM. It implies that the model mutates over time: new topics and edges appear as the new content arrives. If one does not control the process, the model may degrade over time. To avoid this, we have been using a "good edges to all edges" ratio with automatic notification about model's degradation, so that there is no need in human assessment of the model. The fig. 6 shows some examples of "good", "moderate" and "bad" edges according to the EmbedSim metric.

8.2 Improving the quality of already built models

Another example was an improvement of a previously built model which was too large to be rebuild from scratch. We can't usually change the topics in such a situation, but we can change the hierarchical relations between them.

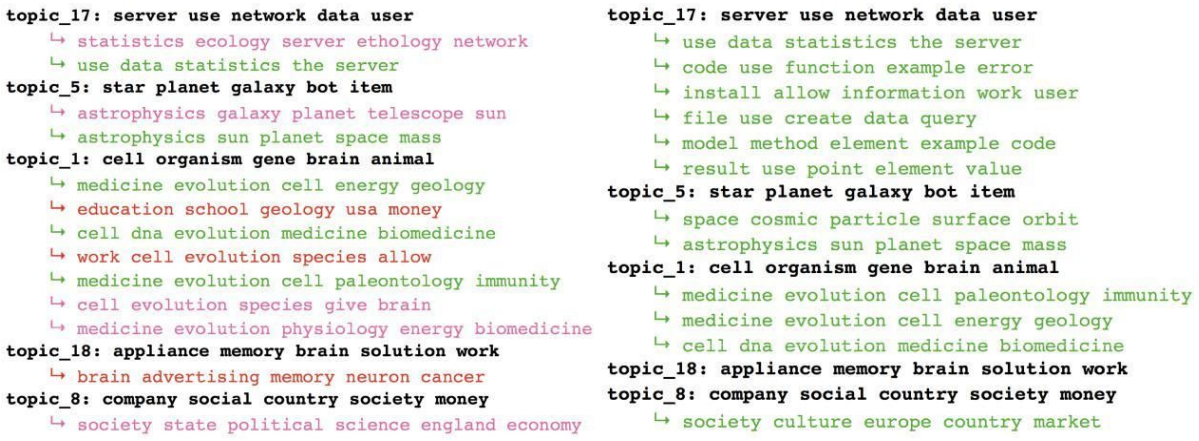

Fig. 7. Hierarchical relations improvement with ranking approach edge selection.

The fig.7 (the left side) demonstrates a subset of the parent topics of the concat model with their child topics. According to our method, we plotted an inverse DP@k graph for the EmbedSim metric of the edges of this model (see section 7.2), found its maximum (in our case it was at k = 100) and built a new hierarchy that contained only the top-k of the edges. The fig. 7 (the right side) demonstrates how quality of the same model increased without rebuilding the model itself. One can see that the new hierarchy looks more consistent and elaborated in comparison with the previous one.

9 Results

In this article we proposed several automated metrics for "parent-child" relations of a topic hierarchy. We showed that the EmbedSim metric based on word embeddings reaches significant consistency with the assessors' judgment on whether the connection between topics exists or not. Other metrics demonstrated moderate yet acceptable consistency and can also be used in conjunction with EmbedSim.

We also proposed two approaches for measuring quality of a hierarchy as a whole. Using metrics of edges' quality we examined averaging and ranking approach to build an aggregated quality measure, and showed that better models reach higher scores in comparison with less elaborated models.

Finally, we demonstrated several applications of the metrics for models' hierarchical relations improvement. For instance, the proposed ranking approach can be used for choosing the optimal set of edges to be included into a hierarchy, as shown in the section 8.

Our work extends existing quality metrics from flat topic models to hierarchical ones which, to the best of our knowledge, hasn't been done before.

10 Discussion

One of the possible extensions of this work is to integrate quality evaluation into the process of building hierarchical models. It can be done in various ways. One option is to build an ARTM regularizer [12], hence the process of constructing a model will try to maximize a certain quality measure during training.

In the section 3 we mention that we used a collection-specific regressor to rank new documents before adding them to the base collection. To make the method applicable to an arbitrary collection we need to replace the regressor with some general approach learned automatically directly from the given collections. One of the possible solutions is, for example, ranking new documents according to their average tf-idf distance to the base collection. Another possible approach is finding p(t|d) vectors of the new documents in the model defined by Φ_0^1 matrix from the previous method iteration (i.e. the

base collection Φ^1) and, then, measuring their KL divergence with the uniform distribution. The lower its KL divergence is the less the distribution of a given new document over the old topics resembles the uniform distribution. If the distribution is close to uniform it means that the document doesn't fit well into the existing model and should be ranked lower and vice versa.

11 Acknowledgements

The work was supported by Government of the Russian Federation (agreement 05.Y09.21.0018) and the Russian Foundation for Basic Research grants 17-07-01536.

12 References

[1] Blei D. M, Gri ths T., Jordan Michael I., Tenenbaum J., (2003), Hierarchical topic models and the nested chinese restaurant process.

[2] Mimno David M., 0010 Wei Li, McCallum Andrew., (2007), Mixtures of hierarchical topics with pachinko allocation, ICML / Eds. Zoubin Ghahramani. Vol. 227. P. 633–640. URL: http: //doi.acm.org/10.1145/1273496.1273576.

[3] Chirkova N. A., Vorontsov K. V., (2016), Additive regularization for hierarchical multimodal topic modeling, Journal of Machine Learning and Data Analysis [Mashynnoe obuchenie i analiz dannyh], Vol.2 № 2 P. 187-201.

[4] Mimno, D., Wallach, H. M., Talley, E., Leenders, M., & McCallum, A., (2011), Optimizing semantic coherence in topic models, In *Proceedings of the conference on empirical methods in natural language processing* (pp. 262-272). Association for Computational Linguistics.

[5] Fang, A., Macdonald, C., Ounis, I., & Habel, P., (2016), Using word embedding to evaluate the coherence of topics from twitter data. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval* (pp. 1057-1060). ACM.

[6] Nikolenko, S. I., (2016), Topic quality metrics based on distributed word representations. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval* (pp. 1029-1032). ACM.

[7] Schütze, H., & Pedersen, J., (1993), A vector model for syntagmatic and paradigmatic relatedness. In *Proceedings of the 9th Annual Conference of the UW Centre for the New OED and Text Research*(pp. 104-113).

[8] Chaney, A. J. B., & Blei, D. M., (2012), Visualizing Topic Models. In *ICWSM*.

[9] Chuang, J., Manning, C. D., & Heer, J. (2012), Termite: Visualization techniques for assessing textual topic models. In *Proceedings of the international working conference on advanced visual interfaces* (pp. 74-77). ACM.

[10] Blei, D. M., & Lafferty, J. D., (2007), A correlated topic model of science. *The Annals of Applied Statistics*, 17-35.

[11] Marchionini, G., (2006), Exploratory search: from finding to understanding. *Communications of the ACM*, *49*(4), 41-46.

[12] Vorontsov, K., Frei, O., Apishev, M., Romov, P., Suvorova, M., & Yanina, A., (2015), Non-Bayesian additive regularization for multimodal topic modeling of large collections. In *Proceedings of the 2015 Workshop on Topic Models: Post-Processing and Applications* (pp. 29-37). ACM.

[13] Kutuzov, A., & Kuzmenko, E. (2016), WebVectors: a toolkit for building web interfaces for vector semantic models. In *International Conference on Analysis of Images, Social Networks and Texts*, pp. 155-161.

[14] Hang Li (2011), A Short Introduction to Learning to Rank. IEICE Trans. Inf. & Syst., Vol.E94–D, No.10, pp. 1854-1862.