Ann. Data. Sci. (2015) 2(1):61–82
DOI 10.1007/s40745-015-0032-1

Refining a Taxonomy by Using Annotated Suffix Trees and Wikipedia Resources

Ekaterina Chernyak[1] · **Boris Mirkin**[1]

Received: 29 September 2014 / Revised: 19 March 2015 / Accepted: 21 March 2015 /
Published online: 2 April 2015

Abstract A step-by-step approach to taxonomy construction is presented. On the first step, the upper layer frame of taxonomy is built manually according to educational materials. On the next steps, the frame is refined at a chosen topic using the Wikipedia category tree and articles, both cleaned of noise. Our main tool in this is a naturally defined string-to-text relevance score, based on annotated suffix trees. The relevance scoring is used at several tasks: (1) cleaning the Wikipedia tree or page set of noise; (2) allocating Wikipedia categories to taxonomy topics; (3) deciding whether an allocated category should be included as a child to the taxonomy topic, etc. The resulting fragment of taxonomy consists of three parts: the manually set upper layer topic, the adopted part of the Wikipedia category tree and Wikipedia articles as leaves. Every leaf is assigned a set of so-called descriptors; these are phrases explaining aspects of the leaf topic. The method is illustrated by its application to two domains in the area of Mathematics: (a) “Probability theory and mathematical statistics”, (b) “Numerical mathematics” (both in Russian).

Keywords Taxonomy refinement · String-to-text relevance · Utilizing Wikipedia · Suffix tree

1 Introduction: Motivation and Background

Taxonomy of concepts in a knowledge domain, or hierarchical ontology, is a popular computational instrument for representation, maintaining and usage of domain

✉ Ekaterina Chernyak
ek.chernyak@gmail.com; echernyak@hse.ru

Boris Mirkin
bmirkin@hse.ru

[1] Higher School of Economics, National Research University, Moscow, Russian Federation

knowledge [1–3]. A taxonomy is a rooted tree formalizing a hierarchy of subjects in an applied domain. Such a tree corresponds to a generalizing relation between the subjects, usually in the form “A is a B” or “A is part of B”. Automation xxxxof taxonomy building is important for further progress in many areas of data analysis and knowledge engineering including computationally text processing and improving information retrieval [1,4,5]. In the authors’ work, domain taxonomies are used to meaningfully map research results to them either to explore research profiles [6] or annotate research papers [7] or measure the level of research results [8].

A definitive taxonomy of the domain of computer science is maintained by the Association for Computer Machinery; the latest version of the ACM computing classification system can be found at [9]. This classification is well balanced so that: (a) its nodes have approximately equal numbers of children, and (b) its branches have approximately equal numbers of layers. However, there are not so many domains for which sound taxonomies are available. For example, when we decided to shift our efforts from the computer science domain to mathematics for the analysis of synopses of courses in mathematics and related subjects in a Russian university, we discovered a rather disappointing picture.

In Russian, the only publicly available taxonomy of mathematics and related domains is the classification for the government-sponsored Abstracting Journal of Mathematics [10] developed back in 1999. This is somewhat outdated and unbalanced. For example, it lacks such topics as “Discrete mathematics”, “Formal concept analysis” and “Mathematical economics”. It has 157 concepts rooted at the topic “Differential equations” and only four topics rooted at “Game theory”. Therefore we thought that we could develop a reasonable taxonomy of mathematics if used instructive materials by the Russian Higher Attestation Commission (HAC). The HAC is a govermental body to supervise the national system of PhD and ScD theses [11]. Its classifications are regularly updated and made publicly available as “passports of specialties”; the list of specialties is revised once in a decade or two. For the case of Mathematics, HAC classification is illustrated in Table 1. As one can see, it covers just two layers of the mathematics domain and one cannot use it in the analysis of a university curriculum, because more layers are needed to reach an adequate degree of granularity of mathematical concepts.

This defines the problem we are going to address as a problem in taxonomy refinement. We start with a manually set an upper part of the taxonomy, a taxonomy frame including the root subject, and then automatically refine leaves of the taxonomy one-by-one. Therefore, given a leaf subject, we need a method that would find appropriately refined concepts and use them to grow the taxonomy. The problem of refinement of taxonomy subjects has received some attention in the literature. A big question arising before any refinement starts is about the sources for generating refined topics. A naive approach is to take a search engine such as Google and run a specially designed query involving the leaf concept under consideratiuon “A”, such as “A consists of…” or “A is a …” [12]. Such a query would lead to a set of concepts that can be considered as potential subtopics for topic A. This works well if the ontology is represented by means of a formal language, such as OWL, by introducing new logical relations [13]. Yet in a less formal context the approach leads to somewhat dubious and messy results.

Table 1 The set of main mathematics divisions according to [11]. One can easily see differences from the divisions in the classification of Mathematics subjects developed by the American Mathematics Society [26]. For example, the field of computer science here is presented with the *Numerical mathematics*, and Combinatorics, with *Discrete mathematics and mathematical cybernetics*

	Mathematics
1	Real-valued, complex valued and functional analysis
2	Differential equations and dynamic systems
3	Mathematical problems in physics
4	Geometry and topology
5	Probability theory and mathematical statistics
6	Mathematical logics, algebra and number theory
7	Numerical mathematics
8	Discrete mathematics and mathematical cybernetics

Next idea is to use a manually designed universal taxonomy such as Wikipedia so that the choice of topics comes from a well defined hierarchical structure openly available in the Internet. Indeed, the idea of using the Wikipedia as a major source of topics for taxonomy building is becoming much popular [12,14–16]. Wikipedia covers many specific knowledge domains and offers a lot of data types, such as unstructured texts, images, the category trees, revision history, redirect pages and links, etc. There are several features making Wikipedia a unique and highly convenient tool for taxonomy building [17]:

- Wikipedia fills the knowledge gap by encoding large amounts of knowledge in an explicit way.
- Wikipedia is a web of interconnected concepts and named entities, thus showing a high degree of ontologization.
- In Wikipedia, the high quality of its ontologized information is ensured by means of collaborative editing, which enables scalable and open knowledge management.
- Thanks to its massive collaborative approach, Wikipedia is able to cover in depth not only domains pertaining to popular culture but also very specialized domains such as Systems Biology or Artificial Intelligence.
- Wikipedia enables a continuously updated content, which is (i) revised to ensure high quality; (ii) kept up-to-date to reflect changes due to recent events.
- Wikipedia is one of the largest multilingual repositories of knowledge ever created.

In papers [12,14–16] different approaches for constructing [14,15] or refining [12,16] ontologies and taxonomies by using Wikipedia article data are presented. In [15,17] the Wikipedia articles are used as a source of topics, in [16] the Wikipedia category tree, in [14] both the articles and the category labels, and in [12] the Wikipedia infoboxes are utilized. This line of research is recently extended to the issue of enhancing the Wikipedia taxonomies by using additional text collections [18] and to building a taxonomy for a text collection by using Wikipedia [19,20].

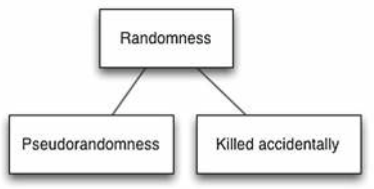

Fig. 1 An example of a wrong subcategory: "Killed accidentally" is a subtopic of "Randomness"

Yet none of this has anything to do with the problem of our concern, refining leaves of a taxonomy with Wikipedia. Nevertheless, in this perspective, using Wikipedia to refine a topic seems a rather straightforward business. First of all, one should find a category in the Wikipedia tree of categories which is nearest to the topic under consideration if it does not coincides with the topic. This can be done by using a topic-to-text relevance measure applied to texts under each category of the Wikipedia tree. Then children of the nearest category are to be considered as the children of the topic, which would complete a step in the refinement process. Thus outlined Wikipedia based refinement strategy will be referred further on as the WR strategy.

Unfortunately, the actual situation with Wikipedia as a crowdsourcing project is a bit messier. One of the issues is that Wikipedia writers sometimes are more enthusiastic than professional. Therefore, one may expect that either the the hierarchy itself or the set of its categories (subjects) or even some articles or all of those may be flawed.

Indeed, the category tree according to the Wikipedia writers is not necessarily a tree. For example, three categories of Wikipedia in Russian, "Optimization", "Machine Learning" and "Search engine" are arranged in such a way that "Machine Learning" is parent of "Optimization" which is parental to "Search engine" which is parental to "Machine Learning". This makes a contour that must be broken, and not necessarily at one edge only.

Next, the category tree is not perfect in the sense that some categories have no semantic relation to their claimed parental categories, the more so with regard to the grandparental categories. An explanation to this phenomenon is given in [21]: Wikipedia writers tend to assign to article or subcategory as many categories as possible. For example, the category "Killed accidentally" lies under the category "Randomness" (see Fig. 1), which is not that bad linguistically speaking. Yet this makes no sense if one wants using that at developing a mathematically oriented taxonomy. Similar examples in the Russian version of the Wikipedia: category "Theory of algorithms" with its subcategory "Feedback loop"; category "Mathematical statistics" with its subcategory "Decision trees"; and category "Algorithm" with its subcategory "Syntactic analysis".

One more source of issues is assignment of articles to categories. Say, in the Russian version of Wikipedia, a stub of article "Percolation theory" is assigned to "Probability theory" category, although it does not properly belong in there; "Artificial life" computational model is assigned to "Evolutionary algorithms"; and article "Linear code" in coding theory is assigned to "Machine learning" category, as well as "Netfix prize" article.

To meaningfully apply the WR strategy, thus, one needs a tool or a set of tools that could be used to evaluate: the similarity between topics, relevance of a category as a subcategory of a topic, relevance of an article to a topic. Using these evaluations one can choose relevant Wikipedia categories and then set thresholds to decide of the relevance of Wikipedia categories or articles to topics depending on the levels of their relevance. To this end, we propose using a naturally defined topic-to-text relevance measure based on building a suffix tree annotated by frequencies to represent the text under consideration as a set of strings consisting of individual letters and symbols. This measure is defined as the conditional probability of characters averaged over fragments of the topic and text being matched (CPAMF) [22–24]. The CPAMF based technique involves no natural language features, which makes it more or less universal across the languages. Moreover, it requires no data preprocessing. On the other hand, the technique has also limitations because it cannot capture the structure of synonyms on its own. In experiments, techniques using suffix tree based relevance measures appear superior over competition [22,25]. For example, [22] reports of a series of experiments in using topics from the ACM Computing Classification System [9] for annotation of research papers according to relevance of the topics to paper abstracts. The CPAMF based relevance measures led to much better results than those based on either of two popular relevance measures, the cosine measure according to the vector space model and the BM25 relevance measure according to a probabilistic model of text [22].

In the remainder, Sect. 2 presents our approach to using the CPAMF based technique to use Wikipedia for refining taxonomy leaves taking into account the noisy structure of Wikipedia. Section 3 describes a version of suffix tree techniques and the CPAMF keyword-to text relevance measure which is used throughout. Two Russian-language examples are given in Sect. 4. Section 5 concludes.

This study (research grant No 15-05-0041) was supported by The National Research University – Higher School of Economics' Academic Fund Program in 2015. The financial support from the Government of the Russian Federation within the framework of the implementation of the 5-100 Programme Roadmap of the National Research University – Higher School of Economics is acknowledged.

2 Our WR Strategy

First we specify the taxonomy domain and manually form the frame of the taxonomy by extracting basic topics from the publicly available instruction materials of the higher attestation commission (HAC) of Russia [27]. The data for refining the taxonomy frame is extracted from Wikipedia. We will provide two examples of refined taxonomies for concepts from: (1) probability theory and mathematical statistics (PTMS) and (2) numerical mathematics (NM). The frames of both taxonomies are three-layer rooted trees of the main topics in the domain (see Tables 2 and 3, correspondingly).

The next step is to define corresponding Wikipedia categories. For each domain we choose only category of the same name, so there is no need to address any other categories. Among the variety of Wikipedia contents we will use only two data types:

- The hierarchical structure of Wikipedia category tree
- The collection of unstructured Wikipedia articles. See Table 4 for the total number of categories and articles.

Table 2 Probability theory and mathematical statistics taxonomy frame

1	Probability theory
1.01	Models and characteristics of random events
1.02	Probability distributions and limit theorems
1.03	Combinatory and geometrical probability problems
1.04	Random processes and fields
1.05	Optimization and algorithmic probability problems
2	Mathematical statistics
2.01	Methods of statistical analysis and inference
2.02	Statistical estimators and estimating parameters
2.03	Test statistics and statistical hypothesis testing
2.04	Time series and random processes
2.05	Machine learning
2.06	Multivariate statistics and data analysis

Table 3 Numerical mathematics taxonomy frame

1	Numerical mathematics
1.01	Algorithms for numerical problem solving
1.02	Numerical method for applied problems
1.03	Software for numerical methods
1.04	Numerical analysis theory
1.04.01	Properties of algorithms
1.04.02	Algorithmic efficiency
1.04.03	Validation of algorithms

Table 4 The total number of subcategories and articles and the number of irrelevant subcategories and articles in PTMS and NM categories in the Russian Wikipedia (accessed in August, 2013)

Domain	#categories	#articles
PTMS	54	928
NM	91	1340
	#Irrelevant categories	#Irrelevant articles
PTMS	20	108
NM	11	30

Hereafter we are going to use the Wikipedia category tree for extending our taxonomy tree. We try to assign some Wikipedia categories to every taxonomy topic of the first and second layer. First, we find those Wikipedia categories that correspond to a taxonomy topic under consideration: they should be subdivisions of the topics. Next we check, whether the assigned category should be further subdivided according to the structure of the category tree. If not, the underlying categories are again assigned to taxonomy topics. Since almost every Wikipedia category contains several articles, the titles of these articles become leaves of our refined taxonomy. Finally, we extract keywords representing the content of each leaf-defining Wikipedia article. These keywords are used then as the leaf descriptors. Since related Wikipedia categories usually

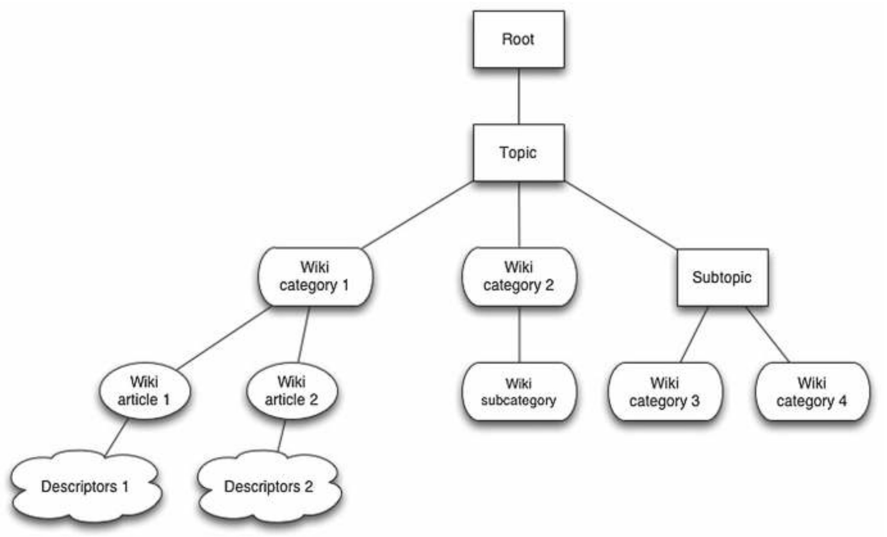

Fig. 2 The refining scheme. Initial taxonomy topics are in *rectangles*, the Wikipedia categories and subcategories are in *rounded rectangles*, the Wikipedia articles are in the *ovals*, and the leaf descriptors are in the *clouds*

have just one- or two-layer subtrees only, such a method seems highly convenient for the task (see Fig. 2 for the refining scheme).

We extract topics from both the Wikipedia category tree and the individual articles. This allows us to follow the above mentioned ACM-CCS golden standard of taxonomy. By restricting the domain of the taxonomy to smaller topics such as the probability theory and mathematical statistics, we avoid the issue of big Wikipedia data and, also, get the possibility to manually examine the results. The method is illustrated by its application to two mathematics domains in Table 1, “Probability theory and mathematical statistics“ and “Numerical mathematics“ (both in Russian), which shows both advantages and drawbacks of the current stage in developing our method.

On the whole, the refined taxonomy should be balanced so that every branch of the taxonomy is approximately of the same depth and width. To achieve that, each topic is refined by one or more layers of Wikipedia categories and articles, placed as leaves at the last layer.

Here are the main steps of our WR approach to taxonomy refining:

1. Specify the domain of taxonomy to be refined and set the frame of taxonomy manually.
2. Download, from the Wikipedia, the category tree and articles from the domain under consideration.
3. Clean the category subtree of irrelevant articles.
4. Clean the category subtree of irrelevant subcategories.
5. Assign the remaining Wikipedia categories to the taxonomy topics.
6. Form the intermediate layers of the taxonomy by using Wikipedia subcategories.
7. Use Wikipedia articles in each of the added category nodes as its leaves.
8. Extract relevant keywords from Wikipedia articles and use them as leaf descriptors.

Table 5 Examples of irrelevant articles in Russian Wikipedia according to the condition A

Domain	Relevance value	Category	Article
PTMS	0.0174	Probability theory	Collectively exhaustive events
PTMS	0.0048	Probability theory	Topic modelling
NM	0.0108	Numerical integration	Verlet integration

Let us describe these steps in more detail using domains of the probability theory and mathematical statistics (PTMS) and numerical mathematics (NM) for illustration.

2.1 Specify the Domain of Taxonomy

As we said already, these are PTMS or NM. See Tables 2 and 3 for the frames of the corresponding taxonomies.

2.2 Download the Category Tree and Articles from the Wikipedia

Download from the Wikipedia the category subtrees, rooted at "Probability theory and mathematical statistics" and "Numerical mathematics" and all the underlying articles.

2.3 Clean the Category Subtree of Irrelevant Articles

We consider that an article in irrelevant to the domain under consideration, if

(a) The relevance score between the article title and the text of the article is low;
(b) The relevance score between the parental category title and the text of the article is low.

The first condition allows us to filter out stubs (short unfinished articles or article templates). According to the second condition we remove those articles that unlikely to have anything to do with the parental categories. The relevance between the title of the parent category and the article is scored by using our string-to-text relevance measure, which follows from the annotated suffix tree (AST) method (described later). It expresses conditional probability of string characters to occur, averaged over the matching fragments in suffix trees, representing a text. It ranges from 0 to 1. The smaller its value, the less is the chance that the string (the title of the parent category) is relevant to the text (the article). We set up the relevance threshold at the value of 0.2 based on our experience in using the measure.

On the first glance, all the judgements of irrelevance in Table 5 seem wrong; yet they are all right. Indeed, the "Collectively exhaustive events" is not an article but just a stub. "Topic modelling" involves probabilities indeed but is part of "Text mining" or "Information retrieval" rather than of "Probability theory". Similarly, "Verlet integration" belongs in "Integration of differential equations" rather than in "Numerical

Table 6 Examples of irrelevant articles in Russian Wikipedia according to the condition B

Domain	Relevance value	Category	Article
PTMS	0.1020	Mathematical statistics	Projection pursuit
PTMS	0.0156	Bayesian statistics	Judea Pearl
NM	0.1948	Regression analysis	ROC curve
NM	0.1944	Numerical integration	BSSN formalism

Table 7 Examples of irrelevant categories in Russian Wikipedia

Domain	Relevance value	Category	Subcategory
PTMS	0.1923	Statistics	State statistics
PTMS	0.1515	Machine learning	Optimization theory
PTMS	0.0142	Statistics	Meta-analysis
NM	0.0632	Algorithms	Computational group theory
NM	0.0287	Numerical methods	Numerical methods for continuum mechanics

integration”. Similar doubts can be raised regarding Table 6 presenting examples of articles irrelevant to their Wikipedia assigned categories according to the condition B above. Yet “BSSN formalism”, as part of the general relativity theory, has nothing to do with “Numerical integration” indeed; the more so that, in fact, it is just a stub, not an article. “ROC curve” is a “Machine learning” concept developed specifically for classifiers, not regression. “Judea Pearl” is not a concept but the name of a renown scientist who has made his name in AI rather than in statistics. Although “Projection pursuit” does belong in “Mathematical statistics”, yet this topic hardly can be considered as an immediate offspring of the “Mathematical statistics” because it clearly belongs in “Multivariate statistics”.

2.4 Clean the Category Subtree of Irrelevant Subcategories

We consider that a subcategory is irrelevant if the CPAMF similarity between its parent category title and the text obtained by merging all the articles in the subcategory is low. The relevance threshold here is set again at the value of 0.2 which probably has something to do with properties of the Russian language.

A few examples of this type are given in Table 7. In one of them, “Optimization theory”, which should be a sibling of “Machine learning”, is assigned as its immediate offspring. The last line relates to a situation in which a rather special branch of computational methods, oriented at a specific domain, comes as an immediate offspring of

Table 8 CPAMF relevance scores between the category "Bayesian statistics" and all the topics in the PTMS fragment of the taxonomy

Relevance score	Topic
0.0190	Time series and random processes
0.0789	Random processes and fields
0.1212	Optimization and algorithmic probability problems
0.1506	Models and characteristics of random events
0.1957	Probability distributions and limit theorems
0.2003	Combinatory and geometrical probability problems
0.2012	Test statistics and statistical hypothesis testing
0.2452	Statistical estimators and estimating parameters
0.2870	Methods of statistical analysis and inference
0.3201	Mathematical statistics
0.3450	Multivariate statistics and data analysis
0.4210	Machine learning
0.5323	**Probability theory**

The category is assigned to the bolded topic

"Numerical methods" in general instead of being classed as belonging to the theory of the specific domain. The other NM example is similar. Two lines in between relate to the meaning of statistics as a social sciences tool and, therefore, do not belong in Mathematics at all.

This approach may fail if the subcategory contains no articles, but is further divided in subcategories, so there is nothing to merge.

2.5 Assign the Wikipedia Categories to the Taxonomy Topics

After clearing the Wikipedia category subtree of irrelevant categories and articles, the method allocates each of the remaining Wikipedia categories to a corresponding topic in the current fragment of taxonomy using the CPAMF relevance scores between the taxonomy topics and the categories. A topic-to-category score is computed between the topic and the text obtained by merging together all the articles in the category, as defined above.

Tables 8 and 9 present two such cases: CPAMF relevance scores between a specific Wikipedia category and all the topics rooted at PTMS (Table 8) and at NM (Table 9). The topics are presented in the order of ascending CPAMF score, so that it is the last one which is assigned to the corresponding category.

2.6 Decision on Wikipedia Subcategories

The categories, which are more relevant to the parental categories than to the taxonomy topic under consideration, remain as intermediate layers in the new taxonomy: their offspring are the relevant articles' titles.

According to the data in Table 10 , the first three subcategories of the category "Random processes", Markov processes, Martingale theory, and Monte Carlo methods, are

Table 9 CPAMF relevance scores between the category "Algorithms for solving SLE" and all the topics in the NM fragment of the taxonomy

Relevance score	Topic
0.1631	Algorithmic efficiency
0.1803	Numerical analysis theory
0.2071	Software for numerical methods
0.2138	Validation of algorithms
0.2761	Properties of algorithms
0.3865	Numerical method for applied problems
0.5134	Numerical mathematics
0.6210	**Algorithms for numerical problem solving**

The category is assigned to the bolded topic

Table 10 Examples of categories, that form intermediate layers

Domain	Relevance to taxonomy topic	Taxonomy topic	Relevance to parental category	Subcategory
PTMS	0.4961	Random processes and fields	0.4842	Stochastic models
PTMS	0.4914	Random processes and fields	0.3825	Noise
PTMS	0.4671	Random processes and fields	0.4813	Markov processes
PTMS	0.4423	Random processes and fields	0.3814	Queueing theory
PTMS	0.4267	Random processes and fields	0.4372	Monte Carlo methods
PTMS	0.3752	Random processes and fields	0.3982	Martingale theory

more relevant to their parent in Wikipedia, rather than to the topic in our tree, whereas the other three are closer to the topic, so that they go immediately under the topic. Therefore we have obtained a subtree in our taxonomy rooted at "Random processes and fields". The root has four children: Random processes, Stochastic models, Queueing theory, Noise. Of these, the first one, Random processes, has three children by itself: Markov processes, Martingale theory, Monte Carlo methods.

2.7 Use Wikipedia Articles in each Added Category Node as its Children

If a Wikipedia category is assigned to a taxonomy topic, all the articles left in it after cleaning are put as new children descending from the topic. For example, the category "Monte Carlo methods" has 10 articles listed in Table 11. As the Table shows, four of the articles are deemed to be irrelevant. Those relevant form the set of children to the category.

The corresponding subtaxonomies are presented on Figs. 3 and 4.

Table 11 Relevant and irrelevant articles to Monte Carlo methods category

Domain	Relevance value	Category	Article
PTMC	0.4529	Monte Carlo methods	Monte Carlo method
PTMS	0.3974	Monte Carlo methods	Monte Carlo method for photon transport
PTMS	0.3864	Monte Carlo methods	Sampling
PTMS	0.3193	Monte Carlo methods	Simulated annealing
PTMS	0.2974	Monte Carlo methods	Gibbs sampling
PTMS	0.2423	Monte Carlo methods	Importance sampling
PTMS	0.1973	Monte Carlo methods	Rejection sampling
PTMS	0.1537	Monte Carlo methods	Slice sampling
PTMS	0.1294	Monte Carlo methods	Fisher Yates shufflel
PTMS	0.0475	Monte Carlo methods	Differential evolution

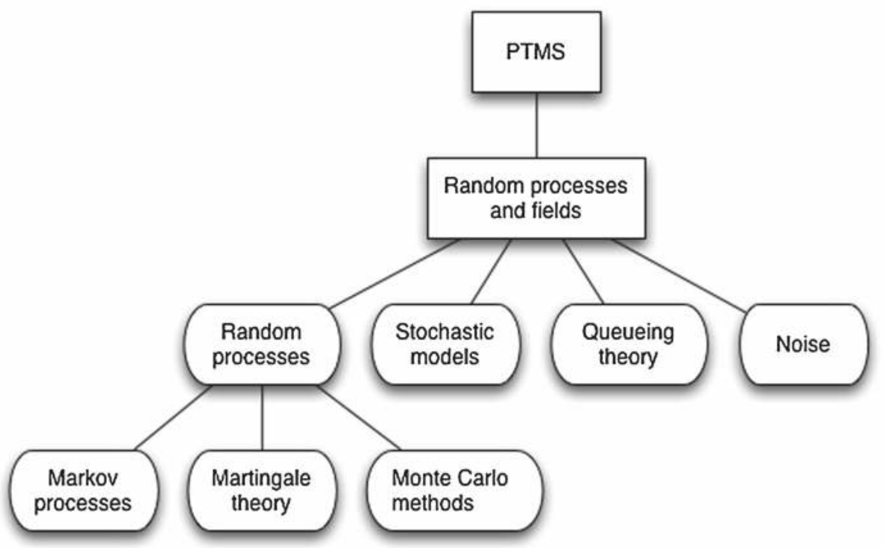

Fig. 3 Random processes and fields subtaxonomy

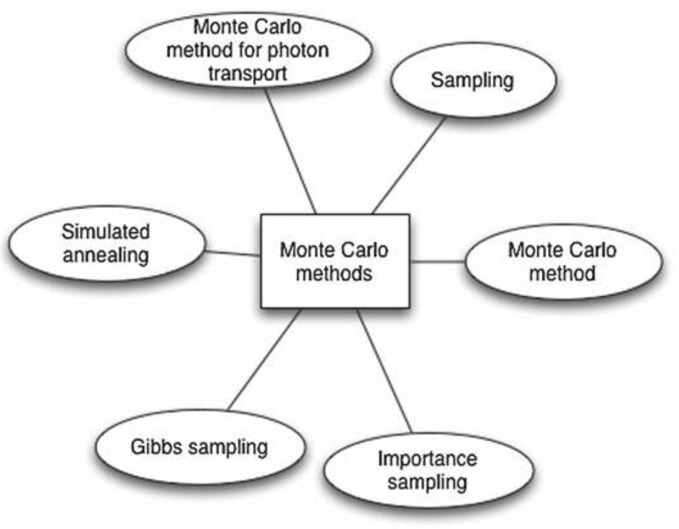

Fig. 4 Monte Carlo methods subtaxonomy

2.8 Extract Keywords from Wikipedia Articles and Use them as Leaf Descriptors

A leaf taxonomy topic can be assigned with a set of phrases falling in it, as is the case of ACM-CCS. To extract keywords and key-phrases, we employ no sophisticated techniques, just taking the most frequent nouns and collocations, respectively. Of course, a key phrase is looked for as a grammar pattern, such as adjective + noun or noun + noun.

More specifically, we use a publicly available part-of-speech parser such as [28] for texts in Russian to label all words in a text by part-of-speech tags. After this we select phrases consisting of neighboring words tagged according to a prespecified pattern like noun + noun or adjective + noun, count the number of their occurrences and select those of the highest frequency. For example, for the leaf “Gibbs sampling” above we received the following most frequent terms and adjective + noun pairs: Table 12.

What is nice about them is that these are exactly terms used in the lecture synopses in Mathematics.

3 CPAMF String-to-Text Relevance Score

The suffix tree is a data structure used for storing of and searching for strings of characters and their fragments [29]. In a sense, the suffix tree model is an alternative to the vector space model (VSM), arguably the most popular model for text representation [30]. When the suffix tree representation is used, the text is considered as a set of strings, where a string may be any semantically significant part of the text, like a word, a phrase or even a whole sentence. An annotated suffix tree (AST) is a suffix tree whose nodes

Table 12 Frequencies of keywords for leaf "Gibbs sampling"

Keyword	Frequency
Random variable	13
algorithm	12
Joint distribution	7
Probability density	6
Conditional probability	4
Deviation	4

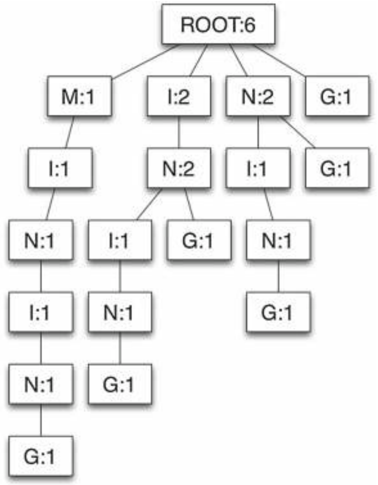

Fig. 5 An AST for string "mining"

(not edges!) are annotated by the frequencies of the strings fragments. An algorithm for the construction and the usage of AST for spam-filtering is described in [25]. Some other applications are described in [23,24].

In our applications we consider a Wikipedia article as a set of its three-word strings. The titles of the Wikipedia categories and articles are also considered as strings in the set. To estimate the relevance of a standalone string to a collection of strings, we build an AST for the set of strings and then find all the matches between the AST and fragments of the given string. For every match we compute the score as the average frequency of a character in it related to the frequency of its prefix. Then the total score is calculated as the average score of all the matches. Obviously, the final value has a flavor of the conditional probability and lies between 0 and 1. In contrast to similarity measures used in [23–25], this one has a natural interpretation and, moreover, does not depend on the text length explicitly nor implicitly, as our experiments show. Let us describe the AST method in more details.

According to the annotated suffix tree model [23–25], a text document is not a set of words or terms, but a set of the so-called strings, the sequences of characters arranged in the same order as they occur in the text. Each string is characterized by a float number. The greater the number is, the more important the string is for the text. An annotated suffix tree (see Fig. 5) is a data structure used for computing and storing all fragments of the text and their frequencies. It is a rooted tree in which:

- Every node corresponds to one character.
- Every node is labeled by the frequency of the text fragment encoded by the path from the root to the node.

To build an AST, we split the text in relatively short strings of three words, and apply them consecutively to warrant that the resulting AST has a relatively modest size. Our algorithm for constructing an AST [25] is a modification of the well-known algorithms for constructing suffix trees [24,29]. The AST is built in an iterative way. For each string, its suffixes are added to the AST one-by-one starting from an empty set representing the root. To add a suffix to the AST, first check, whether there is already a match, that is, a path in the AST that encodes / reads the whole suffix or its prefix. If such a match exists, we add 1 to all the frequencies in the match and append new nodes with frequencies 1 to the last node in the match, if it does not cover the whole suffix. If there is no match, we create a new chain of nodes in the AST from the root with the frequencies 1.

To use an AST to score the string to text relevance we first build an AST for a text in the collection under consideration. Next we match the string to the AST to estimate the CPAMF relevance.

3.1 A Procedure for Computing String-to-Text CPAMF Relevance Score

Input: string and AST for a given text. Output: the CPAMF relevance score.

1. The string is represented by the set of its suffixes; itself included;
2. Every suffix is matched to the AST starting from the root. To estimate the match we use the average conditional probability of the next symbol:

$$score(match(suffix, ast)) = \sum_{node \in match} \phi\left(\frac{\frac{f(node)}{f(parent(node))}}{|suffix|}\right),$$

   where $f(node)$ is the frequency of the matching node, $f(parent(node))$ is it's parent frequency, and $|suffix|$ is the length of the suffix;
3. The relevance of the string is evaluated by averaging the scores of all suffixes:

$$relevance(string, text) = SCORE(string, ast) = \\ = \frac{\sum_{suffix} score(match(suffix, ast))}{|string|},$$

   where $|string|$ is the length of the string.

Note, that "score" is found by applying a scaling function to convert a match score into the relevance evaluation. There are three useful scaling functions, according to experiments in [24] over using a similar method to categorize e-mails in the "spam" and "ham" categories:

- Identity function: $\phi(x) = x$

Table 13 Computing the string "dining" score

Suffix	Match	Score
"dining"	None	0
"ining"	"ining"	$\frac{1/1+1/1+1/2+2/2+2/6}{5} = 0.76$
"ning"	"ning"	$\frac{1/1+1/1+1/2+2/6}{4} = 0.71$
"ing"	"ing"	$\frac{1/2+2/2+2/6}{3} = 0.61$
"ng"	"ng"	$\frac{1/2+2/6}{2} = 0.41$
"g"	"g"	$\frac{1/6}{1} = 0.16$

- Logit function:

$$\phi(x) = \log \frac{x}{1-x} = \log x - \log(1-x)$$

- Root function $\phi(x) = \sqrt{x}$

We use the identity scaling function because it has an obvious meaning: it stands for the conditional probability of characters averaged over matching fragments (CPAMF).

Consider an example to illustrate the described method. Let us construct an AST for the string "mining". This string has six suffixes: "mining", "ining", "ning", "ing", "ng", and "g" . We start with the first suffix and add it to the empty AST as a chain of nodes with the frequencies equal to unity. To add the next suffix, we need to check whether there is any match, i.e. whether there is such a path in the AST starting at its root that encodes / reads a prefix of "mining". Since there is no match between existing nodes and the second suffix, we add it to the root as a chain of nodes with the frequencies equal to unity. We repeat this step until a match is found: a prefix of the fourth suffix "ing" matches the second suffix "ining": two first letters, "in", coincide. Hence we add 1 to the frequency of each of these nodes and add a new child node "g" to the leaf node "n" (see Fig. 5). The next suffix "ng" matches the third suffix and we repeat the same actions: increase the frequency of the matched nodes and add a new child node that does not match. The last suffix does not match any path in the AST, so again we add it to the AST's root as a single node with its frequency equal to unity. Now let us calculate the relevance score for string "dining" using the AST in Fig. 5. There are six suffixes of the string "dining": 'dining", "ining", "ning", "ing", "ng", and "g'. Each of them is aligned with an AST path starting from the root. The scorings of the suffixes are presented in Table 13 .

We have used the identity scaling function to score all six suffixes of the string "dining". Now, to get the final CPAMF relevance value we sum and average them:

$$relevance(dining, mining) = \frac{0 + 0.76 + 0.71 + 0.61 + 0.41 + 0.16}{6} = \\ = \frac{2.65}{6} = 0.44$$

Table 14 The top three candidate taxonomy topics for "Factor analysis" allocation

Domain	Relevance value	Topic	Category
PTMS	0.3700	Multivariate statistics and data analysis	Factor analysis
PTMS	0.3848	Models and characteristics of random events	Factor analysis
PTMS	0.3868	Mathematical statistics	Factor analysis

Table 15 The top three taxonomy topics of the highest score for "Factor analysis" allocation

Domain	Relevance value	Category	Article
PTMS	0.2243	Factor analysis	Principal component analysis
PTMS	0.2563	Factor analysis	Determinacy coefficient
PTMS	0.3587	Factor analysis	Correlation
PTMS	0.5063	Factor analysis	Maximum likelihood
PTMS	0.5337	Factor analysis	Factor analysis

In spite of the fact that "dining" differs from "mining" by just one character, the total score, 0.44, is substantially less than unity. This is not only because the trivial suffix "dining" contributes 0 to the sum, but also because conditional probabilities get smaller for the shorter suffixes. When the similarity is even less noticeable, the score will get even smaller because at the step 2 of CPAMF procedure we divide by the length of the suffix, not the length of the match. This makes the values of the CPAMF score comparable across the strings and texts of various sizes.

4 Results

For the PTMS taxonomy, see Fig. 6, the resulting tree has 6 layers, with its depth varying from 4 to 6. At the cleaning stage 20 categories and 108 articles have been removed from the Wikipedia category tree. The resulting NM taxonomy, see Fig. 7, is of a similar shape: it has 8 layers, the depth varies from 4 to 8. Again at the cleaning stage, 11 categories and 30 articles have been removed.

Now we provide two illustrative examples of how the lower layers of PTMS taxonomy and the higher layers of the NM taxonomy were refined. Specifically, according to Table 14 the category "Factor analysis" should be allocated to taxonomy topic "Mathematical statistics" since it provides the highest score.

There are five articles left in the "Factor analysis" category after the cleaning procedures (see Table 15). The keywords / phrases, extracted from these articles and used as leaf descriptors, are presented on Fig. 6 in clouds.

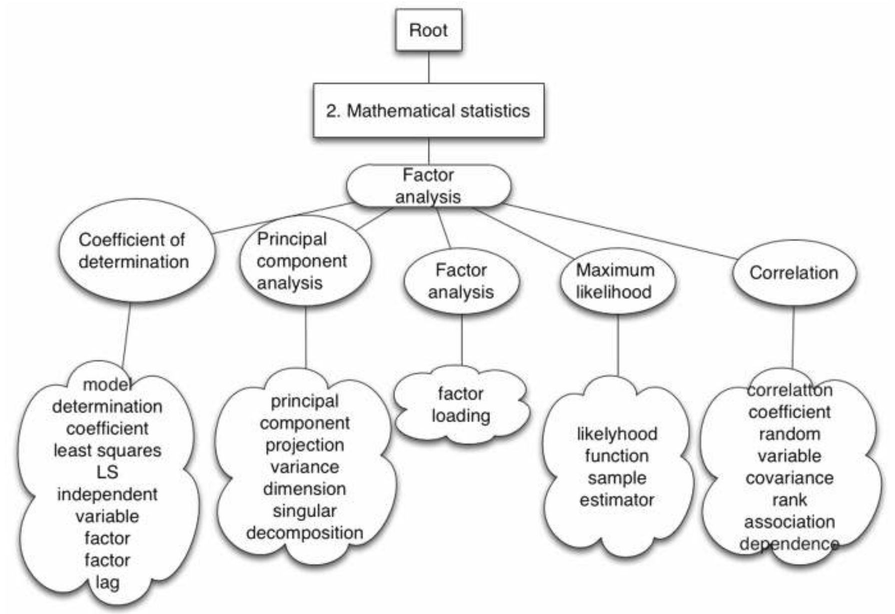
Root
2. Mathematical statistics
Factor analysis
Coefficient of determination
Principal component analysis
Factor analysis
Maximum likelihood
Correlation
model determination coefficient least squares LS independent variable factor factor lag
principal component projection variance dimension singular decomposition
factor loading
likelyhood function sample estimator
correlation coefficient random variable covariance rank association dependence

Fig. 6 A fragment of the refined PTMS taxonomy. Lower layers are shown

There are three categories allocated to taxonomy topic “Numerical algorithms” (see Fig. 7). Two of them (“Optimal control” and “Numerical linear algebra”) contain three articles each, whereas the third one “Numerical integration” contains four articles. The following numbers lead us to this structure of NM taxonomy: see Table 16 for relevance values of category to topic allocation and Table 17 for articles satisfying the cleaning criteria.

There are several issues with each of the obtained taxonomy trees. First, the position of the topic “Decision Trees” is misleading. According to our method, this topic should be placed under “Mathematical statistics” and be, thus, a sibling of the “Machine Learning” topic. The reason is the low relevance of the string “Machine learning” to any of the four articles in the “Decision tree” category. Second, the category “Transformers/Transducers” ([“Preobrazovateli”] in Russian), which is counted as relevant to the parent category “Algorithm efficiency” is further subdivided in “Piezoelectrics”, “Power sources”, “Sound emitters and detectors”. These concepts have nothing to do with algorithms. They appear just because of the double meaning that the category title has in Russian. Third, both taxonomies are stuffed with articles describing personalities, such as “Probability theorists” or “MIPT Lecturers”. Hence more effective cleaning procedures, including filtering of articles according to their types should be developed. Two fragments of refined taxonomies are presented in Figs. 6 and 7. Also, let us recall that the subtree rooted at “Random processes and fields” has been found a bit unbalanced since Wikipedia has had no articles on Random fields.

To refine a taxonomy at a given topic, the AST method works five times in the process:

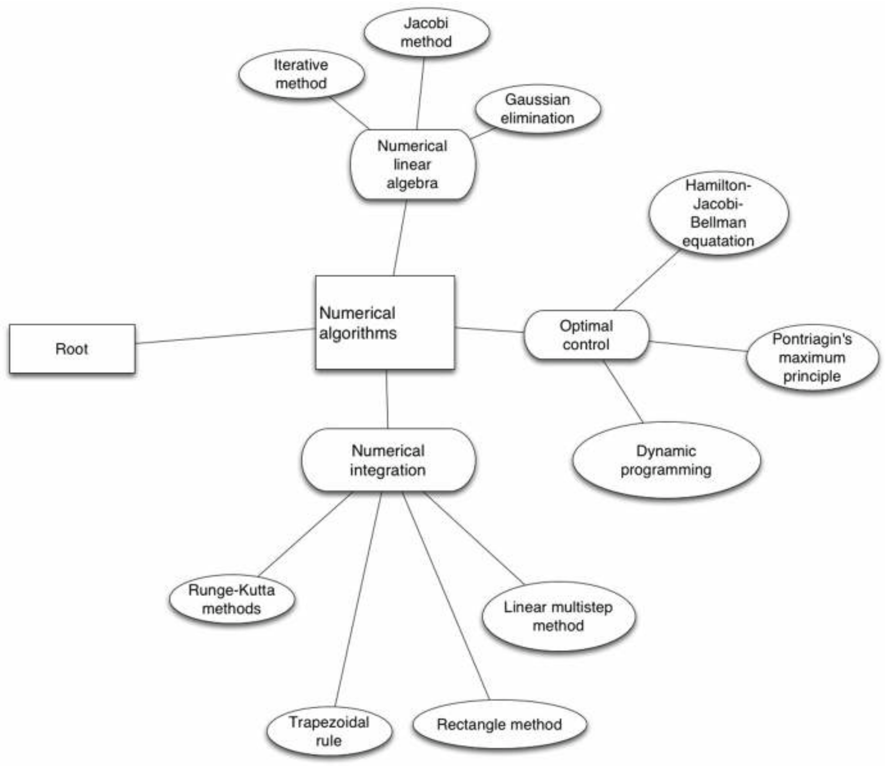

Fig. 7 A fragment of the refined NM taxonomy. Higher layers are shown

Table 16 Three categories allocated to the "Numerical algorithms" topic

Domain	Relevance value	Category	Taxonomy topic
NM	0.4439	Numerical linear algebra	Numerical algorithms
NM	0.4723	Optimal control	Numerical algorithms
NM	0.4877	Numerical integration	Numerical algorithms

- Twice to clean the Wikipedia category subtree of irrelevant articles;
- To clean the category subtree of irrelevant categories;
- To relate taxonomy topics to Wikipedia categories;
- To distinguish between categories to be assigned to taxonomy topics and categories to remain children of their Wikipedia parents.

In the first three cases an "irrelevance" threshold for the article or category title to text should be specified. Our experiments show that the threshold of 0.2, which amounts to 1/3 of the maximum value, works well.

Table 17 Articles relevant to the “Numerical algorithms” branch of NM taxonomy

Domain	Relevance value	Article	Category
NM	0.2819	Iterative method	Numerical linear algebra
NM	0.3642	Jacobi method	Numerical linear algebra
NM	0.3745	Gaussian elimination	Numerical linear algebra
NM	0.4159	Dynamic Programming	Optimal control
NM	0.4423	Hamilton–Jacobi–Bellman equatation	Optimal control
NM	0.7539	Pontriagins̒ maximum principle	Optimal control
NM	0.4321	Runge–Kutta methods	Numerical integration
NM	0.4429	Linear multistep method	Numerical integration
NM	0.4860	Rectangle method	Numerical integration
NM	0.4877	Trapezoidal rule	Numerical integration

5 Conclusion

We have presented an approach at refining a taxonomy by using the Wikipedia and its structure. Our contribution: (a) CPAMF string-to text relevance measure; (b) using CPAMF for cleaning the Wikipedia out of irrelevant categories and articles; (c) using both Wikipedia articles and categories for adding to the topic under consideration two layers at once; (d) supplying the leaves with descriptors. We think that the last item is important as it can be seen as a further refinement of the taxonomy step, so that synopses of university courses can be meaningfully mapped to the taxonomy.

The presented implementation of the approach, by using the CPAMF relevance scores, has both positive and negative sides. The positive relates to a relative independence on the language and its grammar; the negative, with the lack of tools for capturing synonymy and near-synonymy. Other issues can be related to the fact that Wikipedia may give a bit biased picture of the domain. Extension of the method to cover synonymous words with little degree of coincidence should be one of the main subjects for the further work. Another direction for further developments is in developing more precise Wikipedia preprocessing and analysis procedures.

References

1. Snomed ct—systematized nomenclature of medicine clinincal terms (2014) www.ihtsdo.org/snomed-ct/. Accessed 09 Oct 2014

2. Loukachevitch N (2011) Thesauri in information retrieval tasks. MSU, Moscow (In Russian)
3. Robinson P, Bauer S (2011) Introduction to bio-ontologies. Chapman & Hall, London
4. Sadikov E, Madhavan J, Wang L, Halevy A (2008) Clustering query refinements by user intent. In: Proceedings of the 19th international conference on world wide web, pp 841–850
5. White R, Bennett P, Dumais S (2010) Predicting short-term interests using activity-based search contexts. In: Proceedings of 19th ACM conference on information and knowledge management, pp 1009–1018
6. Nascimento S, Fenner T, Felizardo R, Mirkin B, Nascimento S, Fenner T, Felizardo R, Mirkin B (2011) How to visualize a crisp or fuzzy topic set over a taxonomy, vol 6744., Lecture Notes in Computer ScienceSpringer, Heidelberg
7. Chernyak E (2015) An approach to the problem of annotation of research publications. In: Proceedings of the Eighth ACM International Conference on Web Search and Data Mining, WSDM '15. ACM, New York, NY, USA, pp 429–434
8. Orlov M, Mirkin B (2014) Research impact: level of results, citation, merit. Working papers by NRU HSE, Series WP7 ”Mathematical methods for decision making in economics, business and politics”. www.hse.ru/pubs/share/direct/document/140119499
9. ACM computing classification system 2012 (ACM CCS) (2008) www.acm.org/about/class/2012. Accessed 09 Oct 2014
10. Taxonomy of abstracting journal “mathematics” (1999) http://www.viniti.ru/russian/math/files/271.htm. Accessed 09 Oct 2014
11. Higher attestation commission of rf reference (2009) http://vak.ed.gov.ru/ru/help_desk/. Accessed 09 Oct 2014
12. Van Hage W, Katrenko S, Schreiber G (2005) Method to combine linguistic ontology-mapping techniques. In: Proceedings of the 19th International conference on world wide web, pp 34–39
13. Grau B, Parsia B, Sirin E (2004) Working with multiple ontologies on the semantic web. In: Proceedings of the 3rd international semantic web conference, pp 620–634
14. Cui C, Lu Q, Li W, Chen Y (2009) Mining concepts from wikipedia for ontology construction. In: Proceedings of the 2009 IEEE/WIC/ACM international joint conference on web intelligence and intelligent agent technology, vol. 3, pp 287–290
15. Ponzetto S, Strube M (2001) Deriving a large scale taxonomy from wikipedia. In: Proceedings of AAAI conference on artificial intelligence, pp 78–85
16. Wu F, Weld D (2008) Automatically refining wikipedia infobox ontology. In: Proceedings of the 17th international world wide web conference, pp 635–645
17. Hovy E, Navigli R, Ponzetto SP (2013) Collaboratively built semi-structured content and artificial intelligence: the story so far. Artifi Intell 194:2–27
18. Tiziano F, Vannella D, Pasini T, Navigli R (2014) Two is bigger (and better) than one: the wikipedia bitaxonomy project. In: Proceedings of ACL, pp 429–434
19. F-STEP taxonomies (2014) https://sites.google.com/site/focusedtaxonomies/home. Accessed 03 May 2015
20. Medelyan O, Manion S, Broekstra J, Divoli A (2013) Constructing a focused taxonomy from a document collection. The semantic web: semantics and big data. Springer, Heidelberg, pp 367–381
21. Kittur A, Chi E, Suh B (2009) What's in wikipedia? mapping topics and conflict using socially annotated category structure. In: Proceedings of the SIGCHI conference on human factors in computing systems, pp 1509–1512
22. Chernyak E (2015) An approach to the problem of annotation of research publications. In: Proceedings of the eighth ACM international conference on web search and data mining, pp 429–434
23. Chernyak E, Chugunova O, Askarova J, Nascimento S, Mirkin B (2011) Abstracting concepts from text documents by using an ontology. In: Proceedings of the 1st international workshop on concept discovery in unstructured data, pp 21–31
24. Chernyak E, Chugunova O, Mirkin B (2012) Annotated suffix tree method for measuring degree of string to text belongingness. Bus Inform 21(3):31–41 (In Russian)
25. Pampapathi R, Mirkin B, Levene M (2006) A suffix tree approach to anti-spam email filtering. Mach Learn 65(1):309–338
26. Mathematics subject classification (2010) www.ams.org/msc/msc2010.html. Accessed 09 Oct 2014
27. Speciality passports approved by the all-russian higher attestation committee (2014) http://dissertation-info.ru/index.php/2012-08-18-16-13-24/67-2013-01-14-23-56-10.html. Accessed 09 Oct 2014

28. Pymorphy2 part of speech parser (2012) https://pymorphy2.readthedocs.org/en/latest/. Accessed 03 Mar 2015
29. Gusfield D (1997) Algorithms on strings, trees, and sequences: computer science and computational biology. Cambridge University Press, New York
30. Zamir O, Etzioni O (1998) Web document clustering: a feasibility demonstration. In: Proceedings of the 21st annual international ACM SIGIR conference on research and development in information retrieval, pp 46–54