Abstracting concepts from text documents by using an ontology

E. Chernyak[1], O. Chugunova[1], J. Askarova[1], S. Nascimento[2], B. Mirkin[1,3]

[1]Division of Applied Mathematics and Informatics, National Research University Higher School of Economics, Moscow, Russian Federation
[2]Department of Informatics, New University of Lisbon, Caparica, Portugal
[3]Department of Computer Science, Birkbeck University of London, London, UK

Abstract

A method for computationally visualizing and interpreting a text or corpus of texts in a taxonomy of the field is described. The method involves such stages as matching taxonomy topics and text(s) by using annotated suffix trees (ASTs), combining multiple information such as text abstracts, key-words and taxonomy cross-references, building clusters of taxonomy topics and their profiles, and lifting the profiles to higher ranks of the taxonomy hierarchy.

1 Introduction

This paper belongs to a recent trend in the computational ontology-related analysis: usage of ontologies (see, for example, [5], [6], [10], 4]) rather than the development of ontologies, which has been the subject of intense efforts during the past decade. Our ultimate goal is to devise a system that would allow the user to use an ontology of a field for computational interpretation of a text or a set of texts as related to the field. The paper presents some initial stages of our work on the long path towards achieving the goal. This stages include the following: selection of a concept hierarchy (taxonomy) as the formalization of the concept of ontology for the further developments, representation of both texts and taxonomy topics in a unified framework that facilitates and channels sifting the taxonomy topics through the texts to score the matches between them in a comprehensive way, developing quantitative profiles of the texts, clustering them in a way that does not require much input from the user, and, in the very end, lifting the profiles of clusters or individual texts to visualize and interpret them by lifting them to higher ranks of the hierarchy.

As we try to apply the tools developed to Russian sources we face additional issues related to lack of adequate tools for both linguistic analysis and taxonomy development in Cyrillic alphabet.

The following text describes the methodology that is being under both development and, in part, adaptation of a method in [5]. The attempts at applying the methodology to real data are described too. The conclusion underlines the both what has been already made and issues to be tackled.

2 Method's description

2.1 Input

There are two inputs to the method: (1) an ontology and (2) a text collection, both within the same domain.

We consider an ontology to be a rooted tree-like structure of topics, where the parental nodes correspond to more general topics than the children ones. Besides hierarchical relation between topics, other relation might exist. There can be links between topics from different partition.

We explore two such sets: i) the ACM-CCS ontology [1] and a collection of ACM journal abstracts; ii) the VINITI ontology of mathematics and informatics [3] and a collection of teaching syllabuses of mathematics and informatics in NRU HSE (in Russian).

(i) ACM Computing Classification System is used by us as the ontology. The ACM-CCS is a four-level tree that consists of three coded layers and one more, uncoded, layer of topics description. It has eleven major topics on the first llayer such as B. Hardware, C. Computer

Systems Organization, etc. They are subdivided into 81 second-layer topics, which are further divided into third-level topics or so called leave topics. Almost all leave topics are accomplished by topics descriptors that are sets of common phrases or terms corresponding to the topic. There are some cross-references between similar topics in different partitions.

Here is a part of ACM–CCS ontology related to one of its eleven main subjects, D. Software.

- D. Software
  - D.0 GENERAL
  - D.1 PROGRAMMING TECHNIQUES (E)
    - D.1.0 General
    - D.1.1 Applicative (Functional) Programming
    - D.1.2 Automatic Programming (I.2.2)
    - D.1.3 Concurrent Programming
      - Distributed programming
      - Parallel programming

Three coded layers above are presented by topics D., D.0, D.1.0, etc. The topic D.1.3 is supplied with the uncoded topic discription of two terms. The topics D.1 and D.1.2 have references to topics E and I.2.2, respectively.

(ii) The VINITI ontology of mathematics and informatics is the most extensive ontology of mathematics domain in Russian [3]. It is an unbalanced rooted tree of mathematics and informatics topics, provided with a lot of cross-references.

Usually, these ontologies are used to annotate documents or publications in large collections such as ACM portal library or VINITI journals library. Here we concentrate on a different approach to using ontologies – a procedure for abstracting concepts from text documents. Hence, let us consider two collections of texts.

First, we have taken an issue of ACM Journal on Emerging Technologies in Computing Systems (JETC), that is a free access journal [2,8]. Each publication is presented by three items: 1) an abstract; 2) a set of keywords provided by authors; 3) a set of index terms, that are ACM–CCS ontology topics, used on the journal's web site to manually index the article. We use both the abstract and keywords to represent the contents of an article.

Second, NRU HSE teaching syllabuses are involved. These syllabuses correspond to all courses related to Mathematics and/or Informatics as they are taught in the School of Applied Mathematics and Informatics of NRU HSE. They can be easily downloaded from the RU-HSE web-site (http://www.hse.ru).

2.2 Method's composition

The method takes in a text, generates its profile, and then proceeds to further stages described below. A profile is a list of ontology topics generated for an input text. This is based on estimating the degree of similarity between an ontology topic and a text numerically by using the so-called annotated suffix tree (AST) techniques [8].

This is a scheme of the method:

1. texts and ontology preprocessing
2. presenting the texts as annotated suffix trees (ASTs)
3. evaluating similarity between the ontology topics and the texts according to texts' AST features
4. constructing the text profiles
   (a) computing the similarity matrix of the ontology topics according to the text corpus
   (b) computing the similarity matrix between the texts
5. finding and analyzing text clusters
6. finding clusters of ontology topics

7. mapping the clusters into higher layers of the ontology structure.

2.3 Texts preprocessing

The idea is that each text can be split into sentences, while each ontology topic usually consists of one sentence. We represent both the texts and ontology not as symbol sequences but rather as sets of sentences that are taken as strings. To construct a simple machine representation, two stages need to be completed: 1) extracting all the meaningful parts from texts; 2) removing all unnecessary symbols such as html tags, punctuation marks, etc and transforming them to the lower case. While the second step can be easily done automatically, the first one is usually done manually. For example, RU-HSE teaching syllabuses provide not only subjects but some administration issues coverage. The only way to extract a substantial subject description is to do it by hand, since all the teaching syllabuses are formatted, and stored, in various styles and formats.

2.4 Annotated suffix tree representation of a text

An annotated suffix tree (AST) is a data structure, used for computing and representing of all the text fragments' frequencies. AST for a string is a rooted tree, where each node is labeled with one character and one number. Each path from the root to a leaf reads/encodes one of the string suffixes. Frequency of node is the frequency of fragment duplication in given string, that is read/ encoded by corresponding path from the root to a leaf. AST for collection, that is a text, of string reads/ encodes every suffix of each string and their duplication frequency in all strings.
Examination of an ontology and text collection union is done according to procedures, described in [8]. It involves constructing an AST for each text and finding each ontology topic to text relevance evaluation. The simplest way to find an ontology topic evaluation is to construct an AST for the given text and to match all ontology units with this AST. This procedure is more properly described in [8]. Therefore the best ontology topics with the highest estimations for each text might be selected to form the text's profile.

In some cases, a text can be seen as a more complicated entity than just a set of strings. If for example, keywords for a text are provided, one AST may be not enough to represent the keywords-text combination. However ontology, being a hierarchical structure with cross-references, should not be treated as a primitive set of strings too. Here we come to an advanced model of the union.

2.5 Generating profiles: abstracts, keywords, cross-references

Consider a journal publication that is presented as an abstract together with keywords. On the one hand, keywords may be considered as part of the abstract. Hence after building an AST for the strings of the abstract, keywords can be added one by one to the tree. Then the AST is ready to be used for ontology topics evaluation. On the other hand, keywords can be treated apart from abstract as a different constituent of the publication. In this case, one builds two ASTs. First is constructed for strings of abstract, second is constructed for keywords. Thus the process of ontology topics evaluation has to be repeated twice, using both of the created ASTs. These estimations are to be summed, possibly, with different weights, to form the total ontology topic's estimation.

Coming back to the idea of taking cross-reference into account, let us first imagine an ontology as a graph structure. It is composed of two parts: 1) a tree structure, that is hierarchal relation between ontology topics; 2) chaotic references between ontology topics of different levels, that can be interpret as edges of an ontology graph. Hence let us define the distance between two topics as the length of the path between them. In there is no such a path, the distance is set to zero. Now suppose that scores for all ontology topics are computed. We may extend the score of the topic N with the score of the topic N_1 referring to topic N corrected by the distance between N and N_1. For example, let distance between N and N_1 be $distance(N,N_1)$. Hence, the score of the topic N can be set as

$TotalTopicEval(N) := TopicEval(N) + \alpha^{distance(N,N_1)} TopicEval(N_1)$, where α is a fixed constant such as $0 \leq \alpha \leq 1$ and $TopicEval$ is the function of ontology topics scoring.

2.6 Similarity between ontology topics according to the profiles

The process of ontology topics scoring results in topic-text matrix A, where a_{ij} is a total score of the topic i in the text j. In other words, in the matrix A columns are profiles of involved texts. However this matrix can be transformed into similarity matrix of ontology topics by computing dot products of rows of matrix A . This allows us to use spectral clustering methods, which have certain advantages discussed in [11] and [9].

Needles to say is that the following procedures of clustering and cluster lifting expect the similarity matrix (and therefore the clusters) to consist only of leave topics. So from all texts profiles we remove any parent topics to form the similarity matrix.

2.7 Spectral clustering

Additive Fuzzy Spectral Clustering method (FADDIS) [5] applies the Spectral Clustering approach to the Additive Fuzzy Clustering Model. The Spectral Clustering approach relies on the eigenstructure of a similarity matrix. Additive Fuzzy Clustering Model implies finding one cluster at a time by subtracting from the initial similarity matrix those values that correspond to the preceding clusters [5]. Therefore FADDIS method sequentially finds the membership vector of a cluster and its intensity using the maximum eigenvalue and corresponding eigenvector of a matrix, achieved by transformation of the original similarity matrix according to the previously found clusters. Special mention must be made of the pre-processing of the given data: FADDIS involves pseudo-inverse Laplacian transformation of the initial similarity matrix. It was shown by experiments that such a transformation makes the structure of extracted clusters more clear.

2.8 Cluster lifting over ontology

We consider cluster lifting as a way of data aggregation over hierarchically organized ontology. Consequently the idea is to represent a cluster of topics in leaves by a topic that would generalize as many topics in the cluster as possible at a higher level of ontology. From now on we do not need to consider all the elements of a cluster but only the principal subject.

The algorithm [5,6] proceeds according to the assumption that if all or almost all elements of a cluster could be generalized by a topic on a higher layers than the whole cluster “lifts” to that very topic. If the assumption does not hold than lifting is impossible.

To verify this hypothesis we introduce a penalty function to be minimized while lifting the initial set of topics (a cluster) to the root of ontology. It is constructed as weighted sum of different types of vertexes' quantities (numbers). The vertexes are typified during the lifting procedure from leaves to the root. At the level of leaves we have leaves that belong to the cluster considered and those which do not. A topic that generalizes most of the topics in a cluster is algorithmically interpreted at a parent-vertex for most of the vertexes in the cluster assigned the name of head subject. Those vertices that are covered by a head subject but do not belong the cluster are named gaps. And, finally, those vertexes that are not covered by a head subject but do belong to the cluster are called offshoots.

Hereby, we denote H as a number of head subjects, O as a number of offshots and G as a number of gaps, and recursively minimize the function $P = h * H + off * O + g * G$ at each step of lifting (where h, off and g are corresponding weights).

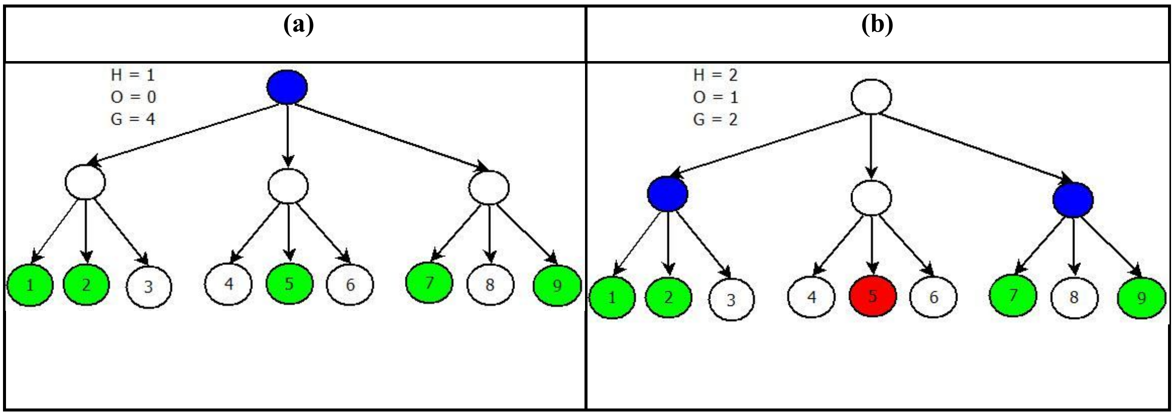

Figure 1. An illustrative example of mapping a subject cluster to an ontology (explained in the text).

Consider the following example of three-layer ontology and a cluster consisting of leaves 1, 2, 5, 7 and 9 (Figure 1). On Fig. 1(a) there is only one head subject that covers leaves 1, 2, 5, 7 and 9 as well as leaves 3, 4, 6, 8 which are gaps. On Fig. 1(b) there are 2 head subjects, one offshoot (leaf 5) and 2 gaps: leaves 3 and 8. The optimal lifting is determined now by the minimal value of penalty in both cases that depends on the relation between the gap and offshoot penalties.

3 Examples of experimental studies

3.1 ACM Journal abstracts

As it was said above, we downloaded and examined a few journal publications. Each of them consists of an abstract, several keywords and indexed terms. Indexed terms are the ACM–CCS ontology topics, that are chosen by publication's authors to characterise it. This supplies a tool for analysis of machine-constructed profiles, based on AST-evaluation of ACM–CCS ontology topics. Yet we are able to find how well the estimated index terms are.

Consider the following example profiles of two journal publications [2, 8], see Tables 1 and 2.

Table 1. Profile A. The spin-wave nanoscale reconfigurable mesh and the labeling problem

AST–profile					ACM index terms
TE	ID	Ontology topic	#	ID	Ontology topic
133.072	C.1.4	Parallel Architectures	1	C.1.4	Parallel Architectures
128.647	C.1.1	Single Data Stream Architectures			
121.059	C.1.2	Multiple Data Stream Architectures (Multiprocessors)			
107.253	D.2.11	Software Architectures	3	C.1.2	Multiple Data Stream Architectures (Multiprocessors)
105.72	C.1.3	Other Architecture Styles			
...	...	...			

Each of the tables consists of two parts. The left part presents our machine generated annotated suffix tree profile (AST–profile), the right one stands for the index terms, which were used by publications' authors to annotate the publication.

The profile A was constructed for the publication that can be previewed on the following web page (http://portal.acm.org/citation.cfm?id=1265951). The profile B was generated for the publication on http://portal.acm.org/citation.cfm?id=1265956 . Only five best scoring ontology topics are present here. However, the AST-profile consists of the all leaves of the ACM CCS ontology.

Table 2. Profile B. A self-organizing defect tolerant SIMD architecture

AST–profile					ACM index terms
TE	ID	Ontology topic	#	ID	Ontology topic
127.503	C.4	PERFORMANCE OF SYSTEMS	40	C.1.2	Multiple Data Stream Architectures (Multiprocessors)
102.03	B.8.1	Reliability, Testing, and Fault-Tolerance			
79.475	B.4.5	Reliability, Testing, and Fault-Tolerance	108	B.4.3	Interconnections (Subsystems)
76.611	B.8.2	Performance Analysis and Design Aids			
			135	B.6.1	Design Styles
72.382	B.3.4	Reliability, Testing, and Fault-Tolerance			
...	...	...			

In the tables: TE is the total score of the ontology topic, ID is the index of the ontology topic. '#" denotes the place of the ontology topic in the descending sorted order of the profile. We expect that index terms are to be high scored according to their abstracts by the AST–procedure and to be placed on the top of AST–profile.

There are two index terms for the publication A. One can see they both among the top five ontology topics, on the first and on the third place correspondingly. The publication B is annotated with three index terms, that are placed on 40th, 108th and 135th places of the AST-profile. While the profile A should be regarded as more or less satisfactory, the profile B is totally inadequate. This difference is caused by the abstracts. The AST–procedure takes into account only matches between the ontology topic's substrings and the abstract's substrings. If no long substrings match, the whole ontology topic will be scored rather low (long enough subsequence is of 5-7 symbols). In the case of the publication B, there are hardly any matches between the index terms and the abstract longer than 3-4 symbols. In contrast, the abstract of the publication A includes whole words of some ontology topics, such as 'parallel' and 'architecture' . What is more, involvement of the word 'architecture' is the reason why so many ontology topics with this word were high scored

The AST–method is able to detect all the fuzzy matches between an ontology topic and a text. From this point of view the topics "Single Data Stream Architectures" and "Multiple Data Stream Architectures (Multiprocessors)" are identical if only substring "Data Stream Architecture" occurs in the text under examination. The small difference in their total scores may be caused simply by the presence of shorter substrings like 'gle' or even 'e'. Here is the main shortcoming of the AST–method. It is not possible to catch an ontology topic in a text if it is formulated by using other words than in the ontology.

3.2 Syllabuses for HSE courses in Applied Mathematics and Informatics: Preliminary Results

The study of the VINITI ontology and the collection of teaching syllabuses showed several shortcomings, both of the ontology and the syllabuses. First of all, after applying the AST–procedure, we derived the topic-topic similarity matrix and extracted crisp clusters by means of the spectral method mentioned above. Almost all crisp clusters contained topics from Topology partition. It means that one or another notion from topology is studied during almost all mathematical courses. But there is no such a subject in the curriculum. Second, as the VININTY ontology has not been updated since 1980's, it was expected that it may have issues in covering more modern topics in mathematics and informatics. With the help of teaching syllabuses we established several nests of topics that should be possibly added to the ontology. For example, the topic ‘Lattices’ is now a leaf in the ontology. According to our results, it should be a parent node with three offsprings: ‘Modular lattices’, ‘Distributive lattices’ and ‘Semimodular lattices’. Third, the ontology has been found of rather imbalanced in the coverage. The profiles of the ‘Differential Equations’ and ‘Mathematical Analysis’ courses according to the ontology are covering all details. This is no wonder because these two cover almost half of the ontology. Yet profiles of less classical subjects such as ‘Game Theory’ or ‘Programming Theory’ are too small and not informative at all. This is caused by the discordance of major ontology partitions sizes. Finally, we thought that the main teaching subjects are named after the first-level or second-level ontology topics. On the contrary, we found that such divisions as ‘Discrete Mathematics’ have not been set among the high-layer ontology topics in the VINITY taxonomy.

4 Conclusion

An idea and some initial stages of a method of abstracting concepts from text documents is presented. It is based on using ontologies as representation of knowledge. We try to simulate the process of abstraction of texts with an ontology in three coherent steps. First, we match ontology topics to all texts and construct texts profiles, by employing text mining techniques. Next step is performed for a corpus of documents: considering leaf topics as the base for abstraction, we find clusters of topics. Finally, we lift clusters to higher layers of the hierarchy to find and visualize head subjects, along with their gaps and offshoots. The head subjects represent the abstraction sought by the method. The method is being developed as an adaptation of the method from [5]. However, a number of novel procedures have been developed in this work as well. Such are using sentence-by-sentence AST modelling, combining different aspects of the texts (such as key-words and cross-referencing) into the scoring system and the like.

Even the first experimental computations lead us to a number of issues that are to be subject of the further developments. The lack of an adequate taxonomy of Mathematics and Informatics in Russian is among them. The AST technology suffers from the effects of repetitive terms such as “architecture”, “method”, “system” and the like acting as noise to raise the similarity scores when not needed. On the other hand, the scores are dropping down when the texts use slightly different terms for the taxonomy topics. This latter aspect should be treated by using neighbors of the taxonomy topics found in texts retrieved by search engines when queried with the topics. We expect that the neighbours would allow not only better scoring in the cases of different terminology, but also would be useful in filling in the gaps generated by lifting the head subjects. The other directions for development would be extension of the concept of ontology from the hierarchy to (semi) lattice structures and finding adequate formalisms for dealing with situations at which there are several ontologies related to the texts.

5 List of references

1. ACM Computing Classification System (1998), http://www.acm.org/about/class/1998 (Cited 9 September 2008)
2. Eshaghian-Wilner M. M., Khitun A., Navab S., Wang K. L.: "The spin-wave nanoscale reconfigurable mesh and the labeling problem". ACM Journal on Emergering Technologies in Computing Systems (JETC) 3(2) (2007)
3. I. Yu. Nikol'skaya, V. M. Yefremenkova: "Mathematics in VINITI RAS: From Abstract Journal to Databases". Scientific and Technical Information Processing 35(3) 128-138 (2008)
4. J. Mercadé, A. Espinosa, J-E. Adsuara, R. Adrados, J. Segura and T. Maes: "Orymold: ontology based gene expression data integration and analysis tool applied to rice", BMC Bioinformatics, 10:158 (2009) doi:10.1186/1471-2105-10-158.
5. Mirkin B., Nascimento S., Fenner T., Pereira L. M.: Fuzzy Thematic Clusters Mapped to Higher Ranks in a Taxonomy. International Journal of Software and Informatics 4(3), 257—275 (2010)
6. Mirkin B., Nascimento S., Pereira L.M.: Cluster-lift method for mapping research activities over a concept tree. Recent Advances in Machine Learning II, 245-247 (2010)
7. Pampapathi R., Mirkin B., Levene M.: A suffix tree approach to anti-spam email filtering. Machine Learning 65(1), 309-338 (2006)
8. Patwardhan J., Dwyer C., Lebeck A. R.: "A self-organizing defect tolerant SIMD architecture". ACM Journal on Emergering Technologies in Computing Systems (JETC) 3(2) (2007)
9. Sato M., Sato Y., Jain L.C.: Fuzzy Clustering Models and Applications. Physics-Verlag (1997). ISBN:3790810266
10. V. Karkaletsis, P. Fragkou, G. Petasis and E. Iosif: Ontology Based Information Extraction from Text, Lecture Notes in Computer Science, V. 6050, Knowledge-Driven Multimedia Information Extraction and Ontology Evolution, 89-109 (2011)
11. Von Luxburg, U.: A tutorial in spectral clustering. Statistics and Computing, 17 (4), 395-416 (2006)