Utilize Probabilistic Topic Models to Enrich Knowledge Bases

Laura Dietz, Avaré Stewart

Fraunhofer Integrated Publication and Information Systems Institute (IPSI)
Dolivostr. 15, 64293 Darmstadt, Germany

{Laura.Dietz, Avare.Stewart}@ipsi.fraunhofer.de
http://www.ipsi.fraunhofer.de

Abstract. In publication driven domains such as the scientific community the availability of topic information in the form of a taxonomy and associated publications is essential. State-of-the-art methods for topic extraction in the Semantic Web community either need high manual effort (e.g. when using categorization) or rely on error prone techniques such as hierarchical clustering.
We present an alternative solution that uses probabilistic topic models, a technique for unsupervised topic extraction based on statistical inference. The topic model can autonomously perform tasks that require massive data processing; such as identifying topics and associations of publications to multiple topics. Only for tasks requiring intellectual activity and for which no reliable automated techniques are available, is the user is asked for assistance.
In this work we explicate how the results of the topic model are stored in a knowledge base for later reuse. It is described how the stored information can be interpreted to provide diagnostic support for the manual topic refinement. We deliniate how the extracted topic information can be exploited in an community service application for the end user.

1 Introduction

Especially in text driven domains such as scientific research communities, it is very important for users to quickly locate publications that relate to their information needs. Studies on researchers [1] indicate that the topic of a publication is the most important criterion for researchers deciding on whether to read a publication or not. Moreover, is it evident that publications typically have more than one topic associated with them and put different weights on each one.

An application that supports scientific reading should thus be able to a) present a list of publications with information about their topic, b) should allow to search for publications that are about a given topic, c) given a publication should make a statement about other topics that are reflected in the publication, and d) provide a topic taxonomy with associated publications.

We define the concept of "topic" and multi-topic documents as follows.

Definition 1. (Topic) *Topics are latent concepts burried in the textual artifacts of a community described by a collection of many terms that co-occur frequently in context.*

For instance "Ontology representation" and "Reasoning languages" are topics in the Semantic Web community. A topic can be described as a collection of many terms that co-occur frequently in context. The collection includes synonyms (e.g. "language" and "speech") as well as combination of terms that resolve ambiguity in polysems (e.g. "spoken", "language" in contrast to "programming", "language").

Definition 2. (Topic Mixture) *A topic mixture is a description which associates a publication with multiple topics; and for each topic t_i a mixture weight, w_i, describing the influence of topic t_i in the publication.*

This work examines techniques for the automatic identification of topics from a corpus of publications as well as calculation of the mixture weights for each publication's topic mixture and stucturing a taxonomy tree. It describes how the results are translated to an RDF representation for storage in a knowledge base, which we refer to as a semantic resource network (SRN).

This work is carried out in the context of VIKEF project, where SRNs [2] are used as a type of knowledge base having an underlying ontology to describe the domain.

The section 2 describes the current state-of-the-art methods for automated topic extraction and discusses assets and drawbacks. Section 3 gives a review on probabilistic topic models which serve as a theoretic foundation for our approach. Section 4 presents the process of enriching an SRN with the results of a probabilistic topic model and discusses how the user can be supported in the manual steps that remain. It describes how the topic information stored in the enriched SRN can be used to improve community services. Section 5 evaluates our approach in comparison to the related work. Conclusions and future work are discussed in the last section.

2 Related Work

State-of-the-art methods for dealing with topics in the Semantic Web community still rely on manually identified topic taxonomies (such as the Semantic Web Topic Hierarchy[1]) and a categorizer that is trained to automatically associate a corpus of documents to the topics. The training of the categorizer requires so-called labeled training data, that is a collection of publications where each is associated with one of the predefined topics. After the training phase, the remaining publications are automatically associated with one of the topics.

Typical categorizers rely on Support Vector Machines [3] and intepret each word in a publication as a dimension in a vector space. During training they identify hyperplanes for each topic that separate the training data that are labeled

[1] https://wiki-sop.inria.fr/wiki/bin/view/Acacia/KnowledgeWeb

with this topic from the data that are labeled with a different topic. After the training, they predict a publication as being about a topic, if its representation in the vectors space lies on the right side of the topic's hyperplane.

The alternative classifyer Cora [4] supports the knowledge engineer in taking a set of (unlabeled) publications, a topic taxonomy and a set of typical keywords for each topic category as inputs. During initialization, each publication that contains one of the keywords is associated with the topic, then the keywords are refined based on statistical inference.

The manual engineering of topics and the creation of labeled training data is difficult and error-prone even for domain experts, because it requires the engineer to already know everything about the domain at hand. Further, this methods suffers from the bootstrap problem. If insights should be gained in a new research area, where a qualified domain expert is absent, it is not possible to engineer the taxonomy nor provide useful training data. Another drawback is that topics in a research community develop over time, thus the taxonomy needs periodical updates and retraining.

An alternative to the manual creation of taxonomies and labeled training data is to apply hierarchical clustering [5]. This relies on a pair-wise similarity measure between the publications. The most commonly used similarity measure is based on the frequency of co-occuring words by using the term frequency inverse document frequency (TF-IDF) measure. The agglomerative hierarchical clustering starts by interpreting each publication as its own cluster and then iteratively merges the pair of mostly similar clusters. Divisive hierarchical clustering on the contrary starts by interpreting all publications as one cluster and then calculates a split between the publications that are least similar to each other. Recent findings in the Machine Learning community [6] indicate that hierarchical clustering does not produce good results. The reason is that on each level of the hierarchy greedy strategies are employed, whose errors propagate to the remaining levels. On the other hand, learning the complete tree of a taxonomy at once [6] is a complex process that needs long run time as well as large amounts of data.

The semi-automatic engineering of topic taxonomies [7] is a third option, where unsupervised machine learning methods are used. Such techniques generate suggestions for topics, associate documents to topics and a taxonomy tree. The knowledge engineer can modify and reject any suggestions made by the system. The knowledge engineer is provided with an application front end for editing the knowledge base, that provides useful background information for each of the tasks. For example, to support assigning documents to topics, the front end displays a similarity plot of documents that are associated with a given topic versus those that are not. The suggestions are calculated via a heuristic combination of Latent Semantic Analysis (LSA) [8], a technique that builds on matrix factorization for identifying latent concepts, K-Means [9], a centroid based clustering algorithm, and Support Vector Machines [3]. The main drawbacks arise from issues of these techniques. For instance, [10] points out that it is questionable whether the results of Latent Semantic Analysis follow the intuition of

topics. It has been proven that K-Means fails to find useful clusterings if data is distributed in certain shapes [11].

We propose to extend the semi-automatic topic extraction approach with a different theoretic underpinning that turned out to provide more reliable results [10] so that user intervention can be reduced. In our case, unsupervised machine learning methods are used for tasks that require massive data processing, such as identifying topics and the association of publications to topic mixtures. Only for tasks that require intellectual activity and where no reliable automated techniques are available, the user is asked for assistance. This is the case in creating pretty labels for the topics and structuring the topics in a taxonomy tree.

Our approach employs probabilistic topics models, which provide a unified view instead of a heuristic combination. Bayesian statistics allows the encoding human intuitions about the properties of topics, for instance that the number of topics associated with a publication should be quite low [12]. Probabilistic topics models are an unsupervised machine learning method, i.e. they do not rely on manually labeled data but are purely driven by the collection of publications. They extract common topics from a given corpus by analyzing the cooccurrence of terms in the documents and are even capable of identifying multiple topics per document, which is explicated in the next section.

3 Probabilistic Topic Models

Probabilistic topic models like probabilistic Latent Semantic Analysis (pLSA) [10] or latent Dirichlet Allocation (LDA) [12] employ bayesian statistics to infer common topics from a corpus of text documents.

As quite common in machine learning and information retrieval methods, topic models process the input data in bag-of-words format [13]. In bag-of-words representation only the frequencies of words in each publication are considered, whereas their order of appearance is ignored. So each document in the corpus is represented by a vector of integers $(f_1, f_2, ..., f_{|\boldsymbol{w}|})$ where f_i is the number of times the word w_i occurs within the document (cf. figure 1a). Each word that occurs at least once in the corpus is assigned a unique index i, where $|\boldsymbol{w}|$ is the size of the vocabulary (i.e. number of different words). Stopword removal and stemming [13] may also be applied during in a preprocessing step, but recent findings [3] indicate that this does not always improve the results.

Probabilistic topic models express the coupling between documents and words by introducing an intermediate layer of hidden (i.e. unobserved) variables that turn out to capture the notion of common topics. Instead of directly associating documents to words, they associate each document with some topics and each topic with some significant words (see figure 1b). Given the bag-of-words representation of the documents and the number of topic variables to identify, the model determines associations from documents to topics and topics to words so that the corpus is represented best.

The association of a document d to its topics is formally defined as a multinomial probability distribution $P(T|d)$ over the random variable $T \in$ topic vari-

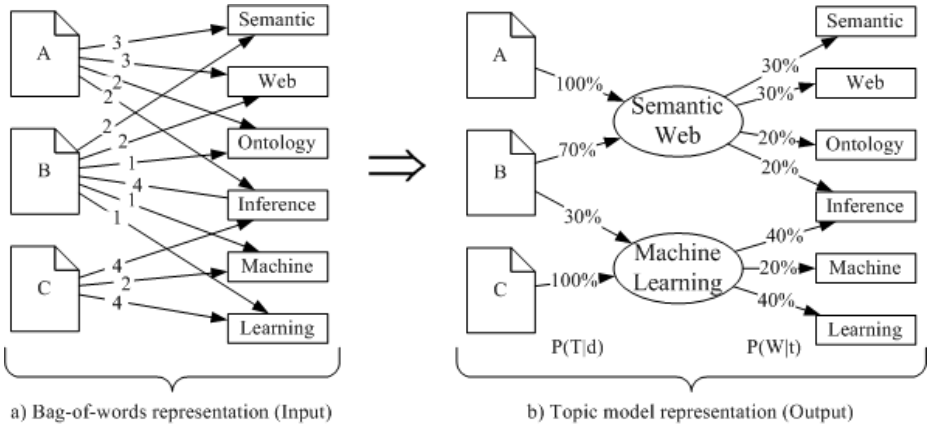

Fig. 1. The given corpus in bag-of-words format (a) is represented by a topic model (b) with mixing proportions of topics in documents $P(T|d)$ and words in topics $P(W|t)$. Document A and C are only about one topic, whereas document B is about both the topics Semantic Web and Machine Learning, with a stonger focus on Semantic Web. The significant words for the both topics are very distinct, but the term "Inference" is used in both topics.

ables conditioned on the document d. For all the possible values of T, i.e. all topic variables, the multinomial probability distribution describes the probability that the document is about this topic. If for a fixed topic t the probability $P(T = t|d)$ is very high, then the topic is considered to be very relevant for the document. Thus, if a document is interpreted as a mixture of topics, these probabilities are the mixture weights as described in definition 2. Since such a proability distribution is extracted for each document d, this leads to a set of distributions for the whole corpus. In the example given in figure 1 the mixing proportions of document B are 70% for the topic Semantic Web and 30% for the topic Machine Learning, where document A is only about Semantic Web. This matches the notion of topic mixtures as introduced in definitions 1 and 2.

Analogously, the association of a topic t to its significant words is defined as a multinomial probability distributions $P(W|t)$. In this case, the possible values of W are words that occur in the corpus (i.e. the vocabulary). And the conditional distribution describes the probability of each word for the given topic t. Note that each topic t is associated with such a distribution, leading to a set of distributions for all identified topic variables.

Definition 3. *(Topic Model) A topic model is represented by*

- *a set of topic variables* $\boldsymbol{t}$ *(Note, that sets of variables are denoted by bold variable names),*
- *a set of probability distributions* $P(T|d)$ *for each document over the random variable* $T \in \boldsymbol{t}$ *that indicate the relevance of each topic for the given document* d*, and*

– *a set of probability distributions $P(W|t)$ for each topic $t \in \boldsymbol{t}$ over the random variable $W \in$ vocabulary that indicates the significance of each word for the given topic.*

During the training phase of the topic model, the topic variables and probability distributions are chosen to represent the given corpus best. This is equivalent to finding a topic model with maximal likelihood $P(corpus|model)$ given the corpus. Let's assume we are given a topic model. The probability that word w_i occurs in document d_i (expressed by token (w_i, d_i)) measures how well this token is represented by the topic model. The probability of the token (w_i, d_i) given the topic model is calculated by equation (1). The likelihood of the model, which measures how well all tokens (w_i, d_i) in the corpus are represented by the model, is calculated by equation (2).

$$P(w_i|d_i, model) = \sum_{t \in \boldsymbol{t}} P(w_i|t) \cdot P(t|d_i) \tag{1}$$

$$P(corpus|model) = \prod_{(w_i,d_i) \in corpus} P(w_i|d_i, model) \tag{2}$$

The task of learning algorithms is to identify the topic model that represents the corpus best by maximizing the likelihood in equation (2). Different learning algorithms for topic models can be found in [14,12,10].

It is also possible to calculate a prediction of $P(T|\tilde{d})$ for an "unseen" document $\tilde{d}$ that was not included in the corpus. This is done by retraining the model with the constraint that topic variables and the assigned significant words are not modified, i.e. that the set of $P(W|t)$ distributions is left unchanged. Since this calculation needs only to determine few variables, the calculation is very fast and can be performed on-the-fly.

4 Incorporating Probabilistic Topic Models in the Semantic Resource Network

A topic model takes a set of publications as input and calculates topic mixtures for each publication as well as a list of significant words for each topic. In this section we describe how the infered knowledge about topics is integrated in the Semantic Resource Network (SRN) and its underlying ontology to calculate, store and query topic information. This lifecycle contains the following steps:

1. Enriching the SRN with results of the probabilistic topic model.
   (a) Querying the SRN to retrieve the publications (or a subset of the publications).
   (b) Applying the probabilistic topic model to preprocessed publications for calculating topic mixtures for each publication.
   (c) Enriching the SRN with topic mixtures and topic concepts.
2. Enriching the SRN with additional topic labels and taxonomy structure.

(a) Creating labels for each of the topics.
(b) Building a taxonomy tree.
(c) Enriching the SRN with labels and taxonomy structure.

3. Exploitation of topic information in the SRN for a community service.
   (a) Querying topic information of for a given publication from the SRN.
   (b) Filtering a list of publications by topic.
   (c) Querying the taxonomy of topic concepts from the SRN.
   (d) Finding relevant topics via sample text.

4.1 Enriching the SRN with results of the probabilistic topic model

Querying the SRN in step 1a requires retrieving all entities of type publication from the SRN and to access their full text. In VIKEF the full text is accessible via the URL stored in the *locationURL* property of the publication entity. The retrieved full texts are converted to the bag-of-words format (with images and equations filtered out) and are used as inputs for the probabilistic topic model in step 1b as described in section 3. In step 1c the topic model is converted to RDF. The OWL schema for encoding topic models in RDF based on the portal ontology[2] presented in figure 2.

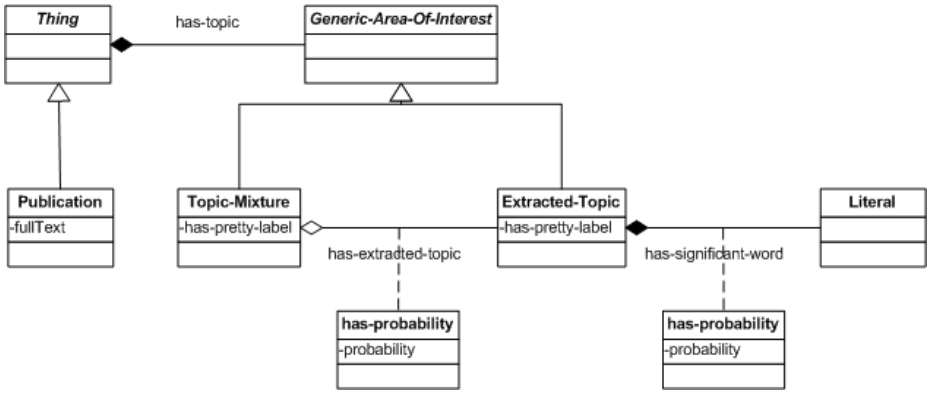

Fig. 2. UML class model on how to encode topic mixtures in RDF based on the portal ontology. Relations used via reification are expressed as association classes.

Each topic variable t is represented by exactly one instance of the entity type *Extracted-Topic*, which is a concrete class of *Generic-Area-Of-Interest* which is used to manually model topic associations. The significant words $\boldsymbol{w}$ for each topic t are associated via the *has-significant-word* relationship. Each of these (t, *has-significant-word*, w) statements gets reified to attach the probability $P(w|t)$ via

[2] http://www.aktors.org/publications/ontology/

the *has-probability* relationship. Note that theoretically all words of the corpus are linked to each topic, but most of them with a very low significance. To ensure expressiveness of the topic information, we suggest to only add an arc in the form of (t, *has-significant-word*, w) if $P(w|t)$ is above a given threshold. Good threshold values are yielded by comparison with the equally distributed case, for instance $P(w|t) > \frac{3}{2} \cdot \frac{1}{|\boldsymbol{w}|}$.

For each publication d, the corresponding publication entity is associated 1:1 with a topic mixture represented by the class *Topic-Mixture* m, which is a subclass of *Generic-Area-Of-Interest* as well. The topic mixture models the association of the publication with several topics t each with the weight $P(t|d)$. Thus, each topic mixture is associated with a number of extracted topics by the *has-extracted-topic* relationship. As above, these statements get reified by the *has-probability* property which stores the $P(t|d)$ values. As with the previous case, the arcs (m, *has-extracted-topic*, t) for topic mixtures m (associated with d) and topics t may be added only if the corresponding probability is above a threshold, i.e $P(t|d) > \frac{3}{2} \cdot \frac{1}{|\boldsymbol{t}|}$.

The probabilistic topic model does not provide pretty human readable labels, but the concatenation of e.g. the five most significant words for each topic is used as a temporary replacement.

Figure 3 gives an example on how the full text that is read from the SRN is used to enrich the SRN with topic information.

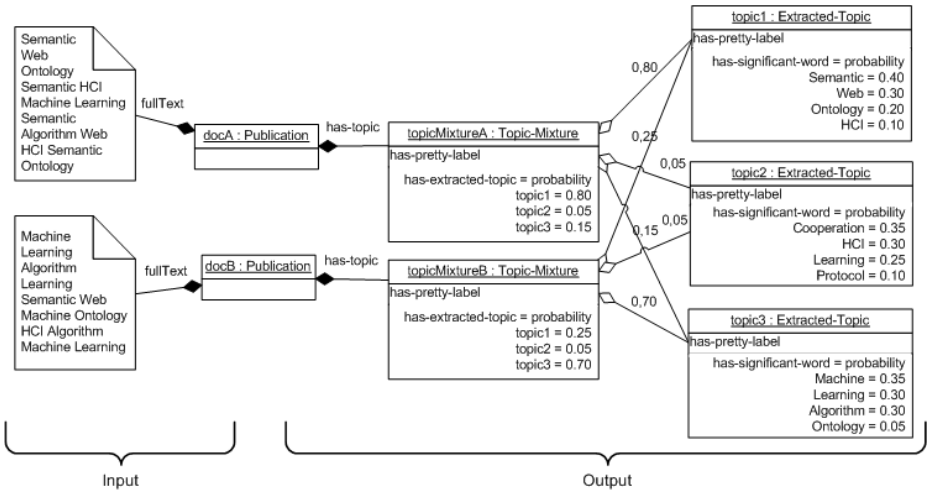

Fig. 3. Example Object UML diagram on how to encode topic mixtures in RDF. The reified has-probability relationship is depicted as a conditional probability table in the attribute section and as labels for the has-extracted-topic associations.

4.2 Enriching the SRN with additional topic labels and taxonomy structure

Since the end user should be provided with a browsable topic taxonomy, expressive labels have to be created for each topic (step 2a) and common abstract topics have to be identified and organized in a taxonomy tree (step 2b).

The temporary replacement of the pretty-labels (consisting of concatenation of most significant words) is probably not what the user expects. For example the temporary pretty-label of topic "Ontology representation" may be "ontology-rdf-syntax-concept-format". Although there are heuristics to extract topic labels from the corpus automatically [15], we suggest to revise them manually to achieve a higher quality. The manual effort is acceptable, since the number of topics is rather low (compared to the number of documents in a corpus). Furthermore, we can support the user in creating the labels with information from the topic model as follows.

For each topic t, the user is provided a list of significant words (taken from the probability distribution $P(W|t)$) and a list of documents that are very relevant for the topic that is given by the distribution $P(D|t)$ which can be computed from $P(T|d)$ via the Bayes Theorem $P(A|B) = P(B|A) \cdot \frac{P(A)}{P(B)}$.

In addition, we can select fragments of documents, that are very much about the given topic t. This is achieved by interpreting each fragment as a new document $\tilde{d}$ on its own and calculating a prediction about its topic mixture as described in section 3 to yield the distribution $P(T|\tilde{d})$. A heuristic for label creation could use this mechanism and return the first noun phrase in such a fragment.

The next step (2b) is to identify common abstract concepts among the topics and organizing them in a taxonomy tree. In the past, hierarchical clustering algorithms have been used to automatically build such a taxonomy [16]. But because of the drawbacks described in section 2, we suggest building the taxonomy manually. This decision is justified by the low number of inner nodes of the taxonomy tree in comparison the the number of topic concepts identified by the topic model. As with the label creation, we can support the user in building the taxonomy by indicating the correlation of topics among documents as well as words. We can derive information about how well the topics are correlated. The correlation of two topics t_1 and t_2 is given by $P(t_1, t_2) \propto \sum_d P(t_1|d) \cdot P(t_2|d)$. In addition, we can present a list of documents that are about both of the topics in contrast to lists of documents, that are only about one of the two.

In order to store the taxonomy tree in the SRN (step 2c) the ontology is extended by the entity type *Compound-Topic* (cf. figure 4). An instance of *Compound-Topic* represents an inner node or root node in the topic taxonomy and aggregates several topic variables (represented by the entity type *Extracted-Topic*) and other inner nodes.

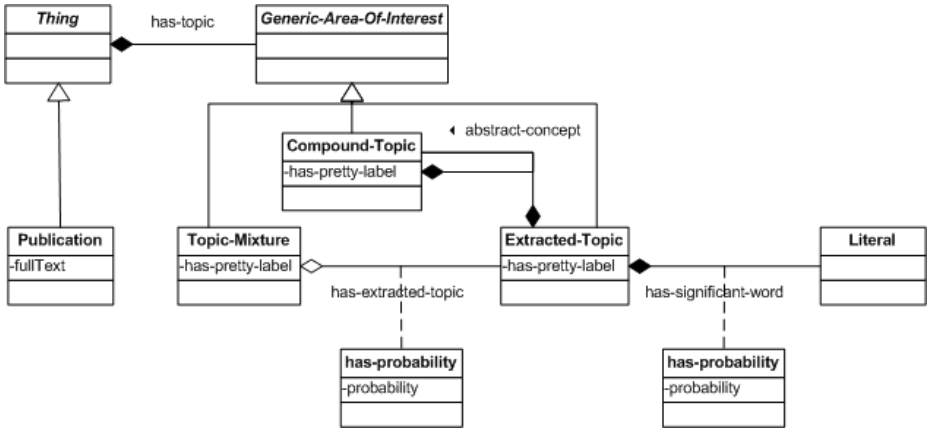

Fig. 4. UML class model that also represents abstract topics as needed in a topic taxonomy.

4.3 Exploitation of topic information in the SRN for a community service

For presentation of topic information in an end user application, the stored topics have to be queried and interpreted appropriately.

One of the simplest cases (step 3a) is to query the topic mixture of a publication which is yielded by following the *has-topic* relation, depicted in figure 4, which will lead to only one *Topic-Mixture* instance. From this topic mixture m the reified *has-probability* statements for all outgoing *has-extracted-topic* relations are examined. They are basically of the form ((m, *has-extracted-topic*, t), *has-probability*, p), where t depicts an instance of *Extracted-Topic* and p is the reified probability. For the given topic mixture m, we could for instance present a bar chart to the end user, where each topic t represents a slice of the size p (cf. figure 5a).

In step 3b the publications are filtered by topic, i.e. only documents that are significantly about a given topic t are presented to the end user. In this case, for the topic t all incoming reified *has-extracted-topic* relations are examined. All publications that are associated with a topic mixture m, where the topic t has a probability value of p which is greater than a threshold like $\frac{3}{2}\frac{p}{number\ of\ topics}$ are listed.

If the user asks for any documents that are significantly about a combination of topics $\bar{t} = (t_1,t_2,..t_k)$, we sum all the probability values $\bar{p} = \sum_{i=1}^{k} p_i$ for each topic mixture m and present the publication that corresponds to m if the summed probability $\bar{p}$ exceeds the threshold. If the user asks for documents of an *Compound-Topic*, this will be interpreted as asking for a combination $\bar{t}$ of all the topics t, that are associated to the *Compound-Topic* via the abstract-concept relation directly as well as transitively.

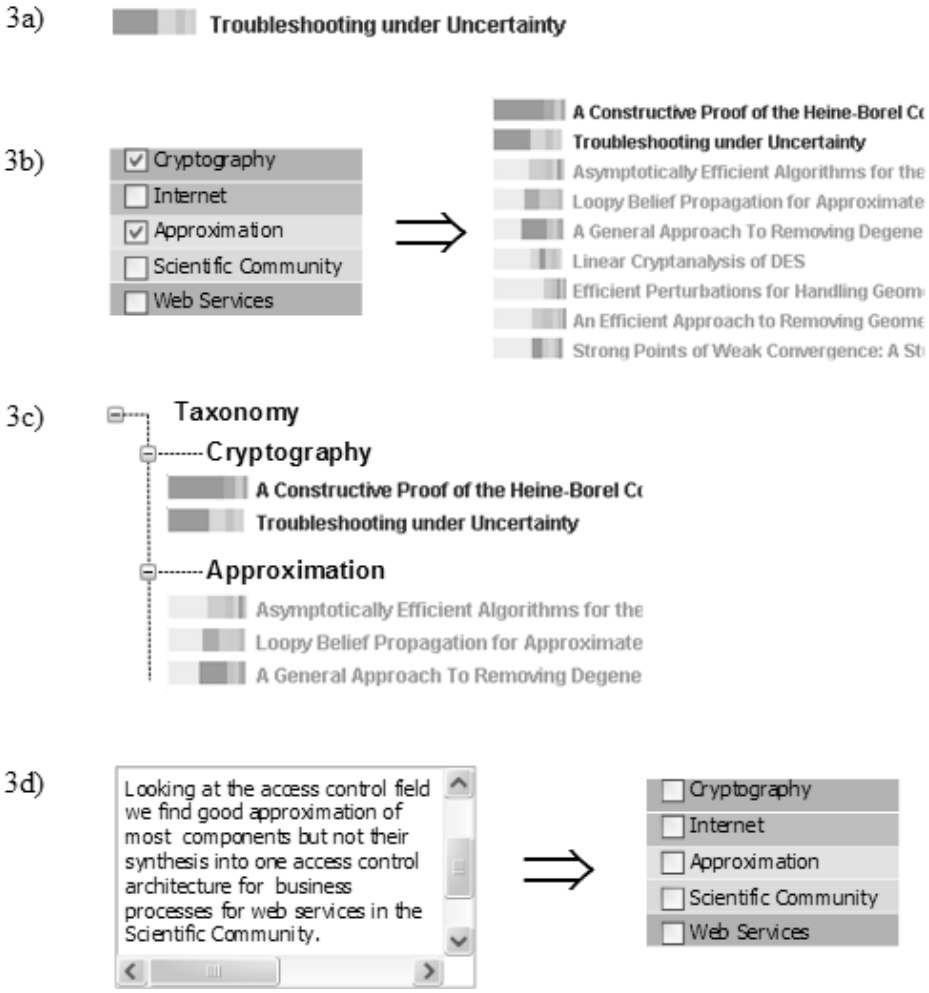

Fig. 5. Exploiting topic information in community services. (3a) Querying topic information of for a given publication from the SRN. (3b) Filtering a list of publications by topic. (3c) Querying the taxonomy of topic concepts from the SRN. (3d) Finding relevant topics via sample text.

In step 3c the taxonomy is queried by retrieving all *Compound-Topic* and *Extracted-Topic* instances and presenting their pretty labels. Each of the nodes in the taxonomy can be used to retrieve a filtered list of documents as described in step 3b. Usually a user expects that each publication is only associated with one leaf in the taxonomy tree, where the filtering approach as described above may associated the document with more than one topic. So we suggest to use the following heuristic instead: A publication with topic mixture m is presented in the category of topic t if the probability p of the *has-extracted-topic* association between m and t is larger than any probability value p' of the other topics t' that are associated with the same topic mixture m.

We can also provide the end user with a search interface, that takes sample text as input and compiles a list of appropriate topics (3d). For this we interpret the sample text as a new publication $\tilde{d}$ and calculate a prediction of a corresponding topic mixture as described in section 3. For that we need to reconstruct the topic model from the SRN. This is possible, since the reified *has-significant-word* relations directly store the $P(W|t)$ values that are needed for the prediction. If relations below a threshold are not stored in the SRN we yield an approximation to the original topic model by interpreting the missing $P(W|t)$ values as zero.

Other topic-based conclusions can be drawn from the enriched SRN. For example, a subset of publications can be displayed according to three topics using a simplex visualization. In addition, Topic Ontology Alignment can be supported matching associated significant words with other topic notions such as controlled vocabulary.

5 Evaluation

We evaluated our approach according to three criteria: manual effort, error proneness and the size of the needed training data. An overview is presented in table 1.

technique	manual effort	error prone	size of training data
manual taxonomy + conventional categorizer	very high	very high	high
manual taxonomy + Cora [4] based categorizer	medium	high	medium
hierarchical clustering [5]	low	very high	medium
learning the taxonomy tree [6]	low	low	very high
semi-automatic topic extraction with LDA [7]	high	low	medium
our approach	medium	low	medium

Table 1. Overview of the evaluation results.

5.1 Manual Effort

The criterion "manual effort" describes how much work is left to the user. This includes the effort in engineering labeled training data, manually determining the structure of the taxonomy or creating human readable pretty labels. The pretty labels have to be manually created in all listed approaches, either as part of a manual engineered taxonomy or after the documents have been associated with the topics.

In addition to the effort for creating the labels, hierarchical clustering and automatic learning of topics and taxonomies require only minimal user involvement in form of an unlabeled corpus of training documents. Cora based categorization needs the specification of the taxonomy with associated keywords in addition to the corpus. In contrast to this, the user involvement is very high when using conventional categorizers, since a reasonable number of documents have to be identified for each topic, and these documents should not address other topics in the taxonomy.

The manual effort of the semi-automatic approach described in [7] is also rated quite high, because the user has to go through all suggestions made by the system and has to correct errors manually. In our approach the manual effort is reduced by employing more reliable topic extraction techniques. The only task left to the user (besides creating the pretty labels) is creating the taxonomy given the correlations of the topics.

5.2 Error Prone

The error proneness criterion summarizes errors arising from the use of error prone technology as well as from very complex tasks which have to be performed by domain experts without any diagnostic support. The error proneness of the technology (as pointed out in section 2) leads to the bad rating of the hierarchical clustering technique.

The conventional categorizer relies on a throuroughly selected set of labeled training data and an appropriate engineered taxonomy. Boths steps have to be performed without any technical support in the general case which leads to the presumption, that this technique should not be recommended to inexperienced users. The Cora categorizer uses the input mainly for bootstrapping purposes and is able to fine-tune itself. Thus it is capable of correcting minor mistakes of the user. Nevertheless we rate this as rather error prone because the topic taxonomy has to be created beforehand without diagnostic support.

The semi-automatic approach of [7] resolves the shortcomings of the employed techniques by compiling a list of suggestions that have to be accepted, rejected or corrected by the user. This way the user gets support in his decisions by the software as well as full control. Our approach builds on more reliably techniques for data intense tasks and follows the approach of [7] for the remaining steps.

5.3 Size of Training Data

The size of the training data indicates how many data must be available to employ a technique. For example the unsupervised learning of topics and taxonomy proposed in [6] requires so many data, because it has to evaluate so many choices (e.g. a tree with n leaves has an exponential number of ways to structure them).

We rate the size of training data needed for a conventional categorizer as high, because it requires a reasonable number of documents that describe each of the topics (without overlap). The training data for the Cora categorizer is smaller, because this needs only a reasonable number of keywords for each topic besides a medium sized (unlabeled) document collection. Approximately the same document collection can also be used as input for hierarchical clustering and the two semi-automatic topic extraction approaches including our approach.

To sum up, the two semi-automatic approaches are most appropriate if the knowledge engineer is rather new to the research domain, i.e. he needs support in identifying the topics, tailoring the taxonomy and associating publications. They require only a medium-sized corpus of publications and thus can be applied also to rather specialized domains such as the proceedings of the ESWC conference series. Our approach provides an improvement by employing a different algorithmic underpinning that uses only one unified model, which provides extracted topics as well as information for supporting the manual steps.

6 Conclusion & Future Work

We presented an approach for automatic extraction of topics from a corpus of publications which also calculates topic mixtures for each publication. This approach can be used as an alternative to other state-of-the-art topic topic extraction methods based on categorizers, where topic labeled publications are not available. In contrast to the categorization approaches, where first a taxonomy has to be engineered, and then publications are assigned to each of the taxonomy leafs, our approach follows the reverse procedure. We suggest to first automatically identify common topics and topic mixtures by using probabilistic topic models and then support the user in generating pretty topic labels and building the taxonomy.

A main contribution of this work is that translation of a probabilistic topic model to an semantic resource network (SRN) by the OWL schema presented in figure 4.

We sketched how a reading support application can exploit the topic information stored in the SRN for providing useful community services.

In the future we will extend the probabilistic topic models to also extract topic mixtures for authors of publications and as well as sub-communities among the authors.

Acknowledgements

The work described in this paper has been partly funded by the European Commission through grant to the project VIKEF under the number IST-507173. We would like to thank the other members of the VIKEF project team for the numerous discussions.

References

1. Meho, L.I., Tibbo, H.R.: Modeling the information-seeking behavior of social scientists: Ellis's study revisited. Journal of the American Society of Information Science and Technology **54**(6) (2003) 570–587
2. Stecher, R., Niederee, C., Bouquet, P., Jacquin, T., Ait-Mokhtar, S., Montemagni, S., Brunelli, R., Demetriou, G.: Enabling a knowledge supply chain: From content resources to ontologies. In: Workshop "Mastering the Gap" on European Semantic Web Conference (ESCW'06). (under submission, 2006)
3. Joachims, T.: Learning to Classify Text Using Support Vector Machines: Methods, Theory and Algorithms. Kluwer Academic Publishers, Norwell, MA, USA (2002)
4. Mccallum, A., Nigam, K., Rennie, J., Seymore, K.: A machine learning approach to building domain-specific search engines. In: Proceedings of the 16th International Joint Conference on Artificial Intelligence. (1999) 662–667
5. Maedche, A., Staab, S.: Ontology learning. In Staab, S., Studer, R., eds.: Handbook on Ontologies. Springer (2004)
6. Blei, D.M., Griffiths, T.L., Jordan, M.I., Tenenbaum, J.B.: Hierarchical topic models and the nested chinese restaurant process. In: Advances in Neural Information Processing Systems. (2004)
7. Fortuna, B., Mladenic, D., Grobelnik, M.: Semi-automatic construction of topic ontology. In: Conference on Data Mining and Data Warehouses (SiKDD 2005). (2005)
8. Deerwester, S.C., Dumais, S.T., Landauer, T.K., Furnas, G.W., Harshman, R.A.: Indexing by latent semantic analysis. Journal of the American Society of Information Science **41**(6) (1990) 391–407
9. Hartigan, J.A., Wong, M.A.: A k-means clustering algorithm. JSTOR: Applied Statistics **28**(1) (1979) 100–108
10. Hofmann, T.: Unsupervised learning by probabilistic latent semantic analysis. Mach. Learn. **42**(1-2) (2001) 177–196
11. Berkhin, P.: Survey of clustering data mining techniques. Technical report (2002)
12. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. Journal of Machine Learning Research **3** (2003) 993–1022
13. Baeza-Yates, R.A., Ribeiro-Neto, B.: Modern Information Retrieval. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA (1999)
14. Steyvers, M., Griffiths, T.: Probabilistic topic models. In Landauer, T., Mcnamara, D., Dennis, S., Kintsch, W., eds.: Latent Semantic Analysis: A Road to Meaning. Laurence Erlbaum (2005)
15. Lawrie, D., , Bruce, W., Rosenberg, A.: Finding topic words for hierarchical summarization. In: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. (2001) 349–357
16. Cimiano, P., Hotho, A., Staab, S.: Comparing conceptual, divise and agglomerative clustering for learning taxonomies from text. In: ECAI. (2004) 435–439