Automatic Labeling Of Topics

Davide Magatti, Silvia Calegari, Davide Ciucci and Fabio Stella
Department of Informatics, Systems and Communications – Universitá degli Studi di Milano-Bicocca
Viale Sarca 336, 20126 Milan, Italy
Email: {magatti, calegari, ciucci, stella}@disco.unimib.it

***Abstract*—An algorithm for the automatic labeling of topics accordingly to a hierarchy is presented. Its main ingredients are a set of similarity measures and a set of topic labeling rules. The labeling rules are specifically designed to find the most agreed labels between the given topic and the hierarchy. The hierarchy is obtained from the Google Directory service, extracted via an ad-hoc developed software procedure and expanded through the use of the OpenOffice English Thesaurus. The performance of the proposed algorithm is investigated by using a document corpus consisting of 33,801 documents and a dictionary consisting of 111,795 words. The results are encouraging, while particularly interesting and significant labeling cases emerged.**

***Index Terms*—Automatic Topic Labeling, Topics Tree , Latent Dirichlet Allocation**

I. Introduction

The problem of topic extraction is attracting a great deal of attention [1], [2], [3] due to its wide applicability; extraction of scientific research topics [4], author-topic analysis [5], opinion extraction [6] and information retrieval [7]. Several probabilistic models have proved to be effective to discover topics. However, the way in which topics are summarized is extremely primitive. Indeed, even if word distributions are intuitively meaningful, it is very difficult to understand what a topic really means and why such a topic is different with respect to other topic. Therefore, the major challenge is to accurately interpret the meaning of each topic. The specialized literature, in absence of an automatic interpretation of the semantics of topic, suggests to either select the most frequent words of the empirical distribution as primitive labels [2], [4], [7], or to manually generate more meaningful labels [6], [8]. Recently, Mei et al. [9], pointed out that neither of the above options is satisfactory. However, in many cases we are in an intermediate situation in which some a priori information is available under the form of topic taxonomy or hierarchy. Significant examples of such a setting are represented by the Medical Subject Headings [10], [11], Google Directory [12], Criminal Law - Lawyer Source [13]. This setting is also typical for businesses and public administrations which over time have built their own categorizations.

In this paper the authors explore this setting and describe the interplay between probabilistic topic models and a priori information for automatic labeling. An algorithm for the automatic labeling of topics is presented. Given a topic, it uses a topics hierarchy, implemented through a tree, to find the *optimal label* according to a set of similarity measures. The most appropriate label is selected by exploiting a set of labeling rules. The approach described in this paper differs from hierarchical clustering [14], [15]. Indeed, it does not build a topics hierarchy to be compared with existing hierarchies (e.g., Open Directory Project (ODP) [16] or a domain ontology) but directly uses ODP for topic labeling.

The rest of the paper is organized as follows: Section II gives basic elements concerning topic extraction. Section III describes the algorithm for automatic labeling of topics. Section IV is devoted to numerical experiments and finally, Section V presents conclusions and discusses further developments.

II. Topic Extraction

Probabilistic Topic Extraction (PTE) is used to analyze the content of documents and the meaning of words, which aims to discover the *topics* mentioned in a document collection. A variety of models have been proposed, described and analyzed in the specialized literature [2], [3], [7]. These models differ from each others in terms of the assumptions they make concerning the data generating process. However, they all share the same rationale, i.e. a document is a mixture of topics.

To describe how the PTE model works, let $P(z)$ be the probability distribution over K topics z, i.e. the *topic distribution*, $P(w|z)$ be the probability distribution over words w given topic z. The *topic-word distribution* $P(w|z)$ specifies the weight to thematically related words.

A document is assumed to be formed as follows: its i^{th} word w_i is generated by first extracting a sample from the *topic distribution* $P(z)$, then sampling a word from the *topic-word distribution* $P(w|z)$. We let $P(z_i = j)$ be the probability that the j^{th} topic was sampled for the i^{th} word token, while $P(w_i|z_i = j)$ is the probability of word w_i under topic j. Therefore, the PTE induces the following probability distribution over words within a document:

$$P(w_i) = \sum_{j=1}^{K} P(w_i|z_i = j) P(z_i = j) .$$

Hofmann [1], [7] proposed the probabilistic Latent Semantic Indexing (pLSI) method which makes no assumptions about how the mixture weights $P(z_i = j)$, are generated. Blei et al. [2] improved the generalizability of this model to new documents. They introduced a Dirichlet prior, with hyperparameter α, on $P(z_i = j)$, thus originating the Latent Dirichlet Allocation (LDA) model. In 2004, Griffiths and Steyvers [4] introduced an extension of the original LDA model which associates a Dirichlet prior, with hyperparameter β, also to

978-0-7695-3872-3/09 $26.00 © 2009 IEEE
DOI 10.1109/ISDA.2009.165

$P(w_i|z_i = j)$. The authors suggested the hyperparameter to be interpreted as the prior observation count on the number of times words are sampled from a topic before any word from the corpus is observed. This choice can smooth the topic-word distribution in every topic, with the amount of smoothing determined by β. Topic extraction, i.e. estimation of the topic-word distributions and topic distributions for each document, can be implemented through different algorithms. Hofmann [7] used a direct estimation approach based on the Expectation-Maximization (EM) algorithm. However, such an approach suffers from problems involving local maxima of the likelihood function. A better alternative has been proposed by Blei et al. [2] which directly estimates the posterior over z given the observed words w. However, many text collections contain millions of word tokens and thus the estimation of the posterior requires the adoption of efficient procedures. Gibbs sampling, a form of Markov Chain Monte Carlo (MCMC), is easy to implement and provides a relatively efficient method of extracting topics from a large document corpus.

III. Topics Tree and Automatic Labeling

The document labeling problem can be tackled in two different ways: human labeling and computer labeling. The first approach maps a document into a set of pre-specified categories, usually such categories form a taxonomy or a topic hierarchy. The latter approach has recently emerged to be well suited, i.e. to be efficient and effective, in several settings. While human labeling usually benefits from the availability of a domain specific topic hierarchy, agreed by experts, it is extremely time consuming and in some particular situations universally agreed labeling cannot be achieved. On the contrary, computer labeling is economically attractive, while the achieved labeling must be accurately checked by specific domain experts to ensure that it is consistent. Furthermore, effective methods for automatically building a topics hierarchy have been very recently proposed in the specialized literature. However, businesses, public administrations, healthcare institutions and lawyers, to mention just a few, have built over time their own domain specific taxonomies and labeling schema. Therefore, they are reluctant to abandon the results achieved over many years of study and analysis. In such cases the solution to the document labeling problem is placed in between the two alternative solutions - namely human and computer labeling. Indeed, we want to maintain our own labeling setting, while at the same performing computer labeling to reduce the time required for document labeling.

The approach we propose offers an efficient answer to these requirements. It implements an intermediate solution, between human and computer labeling, which exploits the available labeling schema to automatically label a document corpus. In more detail, it is assumed that the available labeling schema is summarized through a topics tree whose main features are described in the next subsection.

A. Topics Tree

A *topics tree* is a pair $\Upsilon = \langle V, E \rangle$, where V is a set of nodes indexed by non negative integers $j = 0, 1, ..., N$, while $E = V \times V$ is a set of arcs (i, j) between nodes, $i,\ j \in V$. Each node j is associated with a topic $T_{\Upsilon}(j) = \langle label, words\ list, infos \rangle$, where *label* is the topic label, *words list* is the topic list of positive words and *infos* is additional information associated with the topic. It is worthwhile to mention that the *root node* indexed by 0 is not a proper topic. It is introduced to ensure that the set of topics, which usually gives rise to a forest, forms a tree. Therefore, the *root node* can be interpreted as the most generic topic or *all-the-topics*. It is worthwhile to mention that the framework considered in this paper assumes that the world is described by a set of concepts (equivalently, topics) which are inserted into a light ontology [17]. The *topics tree* Υ describes how topics are linked in a taxonomic way by means of the usual IS-A relation. A concept c IS-A concept d iff $I(c) \subseteq I(d)$, where I is an interpretation function $I : \mathcal{C} \mapsto U$ mapping a concept $c \in \mathcal{C}$ to a subset $I(c)$ of a given universe U. For instance, under the common-sense interpretation, *cat* IS-A *feline* since any real cat belongs to the set of felines (but not viceversa).

B. Similarity Measures

A central element of the algorithm for automatic labeling of topics is the similarity measure used for topics comparison. A large variety of similarity measures has been proposed in the specialized literature. In [18] similarity measures belonging to a representative set are compared and partitioned into two classes. The first class contains all those similarity measures showing a coherent behavior with respect to the semantics of the compared concepts. The second class contains similarity measures which do not show a coherent behavior, i.e. Euclidean similarity, T-similarity, L-similarity and W-similarity. In this paper, we consider a subset of the first class: cosine similarity, overlap similarity, mutual similarity and dice similarity, plus the Tanimoto and the Jaccard similarities. The definitions of the above similarity measures are provided for reason of clarity. Let $\underline{x}$ and $\underline{y}$ be two vectors, $||\underline{x}||$ be the Euclidean norm of vector $\underline{x}$, then the *cosine similarity* between vector $\underline{x}$ and vector $\underline{y}$ is defined as $Cosine(\underline{x}, \underline{y}) := \underline{x} \cdot \underline{y} / ||\underline{x}|| \cdot ||\underline{y}||$[1]. The *overlap similarity* measure is defined as: $Overlap(A, B) := |A \cap B| / min(|A|, |B|)$, where A and B are sets, while $|A|$ represents the cardinality of set A. The *mutual similarity* uses the degree of inclusion of set A into set B and the degree of inclusion of set B into set A and computes their average value, it is defined as follows: $Mutual(A, B) := (\frac{|A \cap B|}{|A|} + \frac{|A \cap B|}{|B|})/2$. The *dice similarity* is defined as follows: $Dice(A, B) := (2|A \cap B|)/(|A| + |B|)$. It is related to the Jaccard coefficient, commonly used in information retrieval to measure the overlap between two sets. The *Jaccard coefficient* is defined as $Jaccard(A, B) := |A \cap B| / |A \cup B|$. Finally, the *Tanimoto similarity* is defined as the complement of Tanimoto

[1] Note that vectors $\underline{x}$ and $\underline{y}$ are binary representations of sets A and B. The dimensionality of $\underline{x}$ and $\underline{y}$ equals the cardinality of the union set $A \cup B$.

metric and is commonly used to compute the distance between sets which have different cardinality. $Tanimoto(A, B) := 1 - ((|A| + |B| - 2|A \cap B|)/(|A| + |B| - |A \cap B|))$.

C. The ALOT Algorithm

An *extracted topic* is a word list, obtained from the application of the LDA method described in Section II. Given a topics tree Υ and a set of extracted topic $\top = \{T_e(1), ..., T_e(K)\}$, the algorithm for Automatic Labeling Of Topics (ALOT) aims to label each element $T_e(i)$, $i = 1, ..., K$, by means of labels associated with topics $T_\Upsilon(j)$, $j = 1, ..., N$ of the topics tree Υ. The main components of ALOT are the similarity measures and the labeling rules. While similarity measures, introduced in subsection III-B, are concerned with the *word list* component of topics, labeling rules exploit the topics tree to find the *optimal label* (w.r.t. the available topics tree Υ) for each extracted topic $T_e(i)$. More in detail, given a topics tree Υ, for each extracted topic $T_e(i)$ its nearest topic $T_\Upsilon(j_r^*)$, with respect to similarity measure S_r, is recovered by solving the following optimization problem:

$$j_r^* = \arg\max_j S_r(T_e(i), T_\Upsilon(j)).$$

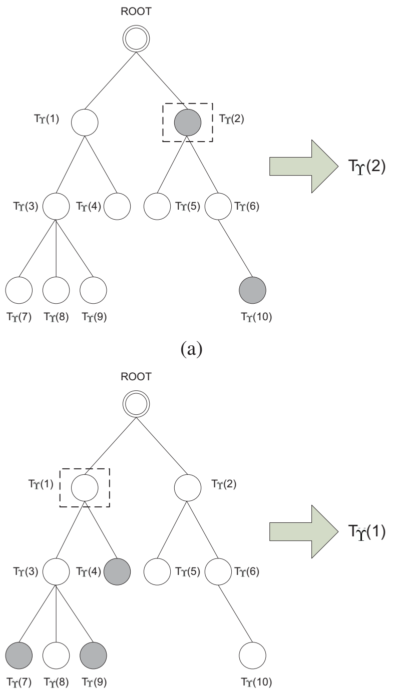
That is, $T_\Upsilon(j_r^*)$ is the topic which has the greatest similarity S_r with $T_e(i)$ and j_r^* the index of this topic in Υ. For each extracted topic $T_e(i)$, we collect all these indexes associated to the six similarity measures S_r introduced in subsection III-B in $L(i) = \{j_1^*, ..., j_6^*\}$, and the corresponding set of topics will be denoted by $\Delta(i) = \{T_\Upsilon(j_1^*), ..., T_\Upsilon(j_6^*)\}$. Both $L(i)$ and $\Delta(i)$ will be represented on the tree structure by coloring the corresponding nodes. For instance, in Figure 1(b), $L(i) = \{4, 7, 9\}$ (and this means that the six similarity measures give only three different results). Given $T_e(i) \in \top$, the following cases can occur:

- *Topic concordance* (**TC**); $j^* = j_1^* = ... = j_6^*$, all similarity measures agree on which the nearest topic $T_\Upsilon(j^*)$ is. The ALOT algorithm labels $T_e(i)$ with the label of the corresponding optimal unique topic $T_\Upsilon(j^*)$.
- *Topic discordance* (**TD**); $\exists l, g : j_l^* \neq j_g^*$, at least two similarity measures disagree on which the nearest topic is. The ALOT algorithm labels $T_e(i)$ according to:
  1) **SA** (Semantic Association, topics belonging to $\Delta(i)$ share a predecessor, different from the root). The ALOT algorithm looks for a topic $T_\Upsilon(j)$, not necessarily belonging to $\Delta(i)$, which synthesizes all the topics in $\Delta(i)$. The following subcases can occur:
     a) **Path**; all the topics in $\Delta(i)$ lie on the same path. ALOT labels $T_e(i)$ with the label of the shallowest topic in $\Delta(i)$, i.e., the topic $T_\Upsilon(j_r^*)$ which minimizes $depth(j_r^*)$ (Figure 1(a)).
     b) **Subtree**; all the topics in $\Delta(i)$ belong to a common subtree. The ALOT algorithm labels $T_e(i)$ with the label of the topic $T_\Upsilon(j)$ which is the common deepest predecessor of topics in $\Delta(i)$ (Figure 1(b)). $T_\Upsilon(1)$. Notice that the case where $T_\Upsilon(j) \in \Delta(i)$ can also occur.

Algorithm 1 Automatic Labeling Of Topics (ALOT)

```
Require: A topics tree $\Upsilon$, a topic $T_e(i)$ to be labeled.
Ensure: The label of $T_e(i)$.
Compute $j_r^* = \arg\max_j S_r(T_e(i), T_\Upsilon(j)) \; \forall r, r = 1, ..., 6$,
    and set $L(i) = \{j_1^*, ..., j_6^*\}$
if $j_1^* = ... = j_6^*$ then {case TC}
    Return the $T_\Upsilon(j_1^*)$ label
else {case TD}
    Case Path: Find $j$, the swallowest topic in $\Delta(i)$
    Case Subtree: Find $j$, the deepest predecessor of nodes
        belonging to $\Delta(i)$
    if $T_\Upsilon(j) \neq ROOT$ then {case SA}
        Return the $T_\Upsilon(j)$ label
    else {case NSA}
        Compute $j^{max}$ which maximizes $depth(j^{max})$ and
            $|successor(j^{max}) \cap \Delta(i)|$
        if $j^{max}$ is unique then {case S-dtmap}
            Return the $T_\Upsilon(j^{max})$ label
        else
            Apply subcase M-dmatp and return the computed
                label if unique or ROOT if not (subcase R-dmatp)
        end if
    end if
end if
```

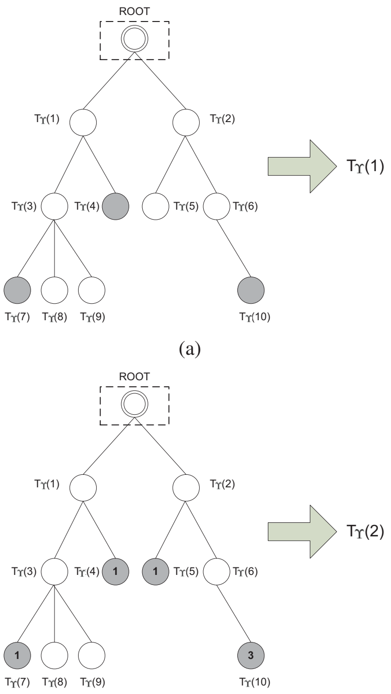
  2) **NSA** (Non-Semantic Association, topics belonging to $\Delta(i)$ do not share a predecessor, except from the root). ALOT uses a majority voting scheme and selects the deepest maximally agreed topics predecessor, i.e. the topic $T_\Upsilon(j^{max})$ associated with the node j^{max} such that $depth(j^{max})$ and $|successors(j^{max}) \cap L(i)|$ are both maximized. The following subcases can occur:
     a) **S-dmatp** (Single deepest maximally agreed topic predecessor); a single topic $T_\Upsilon(j^{max})$ is obtained and its label is associated with $T_e(i)$. In Figure 2(a), the selected topic is $T_\Upsilon(1)$ since it has two successors in $\Delta(i)$ compared to only one on the other branch of the tree.
     b) **M-dmatp** (Multiple deepest maximally agreed topic predecessor); more than one topic is returned by the majority voting scheme. ALOT computes how many times each $T_\Upsilon(j_l^*)$ is a descendant of all the $T_\Upsilon(j^{max})$, stores this information into *info* and finds the $T_\Upsilon(j^{max})$ with the maximum number of occurrences. In Figure 2(b), the majority voting returns $T_\Upsilon(1)$ and $T_\Upsilon(2)$ and between them, $T_\Upsilon(2)$ is selected since it has four successors in $\Delta(i)$ compared to only two of $T_1(x)$.
     c) **R-dmatp** (Rooted deepest maximally not agreed topics predecessor); the $root\ node$ is returned by the majority voting scheme and the maximum number of occurrences $T_\Upsilon(j^{max})$ is the same



(a)

(b)

Fig. 1. SA cases: (a) Path and (b) Subtree.



(a)

(b)

Fig. 2. NSA cases: (a) S-dmatp and (b) M-dmatp.

for at least two descendants in $\Delta(i)$. Then, the $root\ node$ is returned by ALOT.

IV. Numerical Experiments

The document corpus has been obtained by exploiting the *Google Directory* (gDir) which relies on ODP. This project manages the largest human-edited directory available on the web. Editors guarantee fairness and correctness of the directory. gDir is a topics tree, organized into 16 topics, where each node is associated with a topic while the labels of its children form its word list. The tree is unbalanced, some branches have small depth (`News`) whereas others have large depth (`World`).

A. Document Corpus and Text Preprocessing

The corpus has been generated by submitting a set consisting of 960 queries to the Google search engine through the Google Ajax API. Each query is formed by a couple of words randomly selected from the union of word lists associated with the topics tree. Some examples of random queries are `''Music Environment''`, `''News and Media Current Events''`,`''Holidays Ukrainian''`. For matters of simplicity, the results are filtered and only PDF files written in English are retrieved. The query process retrieved 46,480 documents. The document corpus has been preprocessed by means of plain text transformation, stopwords removal and size-based filtering: only documents with a size between 2 and 400 KB have been retained. The filtered document corpus consists of 33,801 documents while the global vocabulary, consisting of 111,795 words, has been obtained by filtering out those words mentioned in less than 10 or in more than 2,551 documents.

B. Topic Extraction and Selection

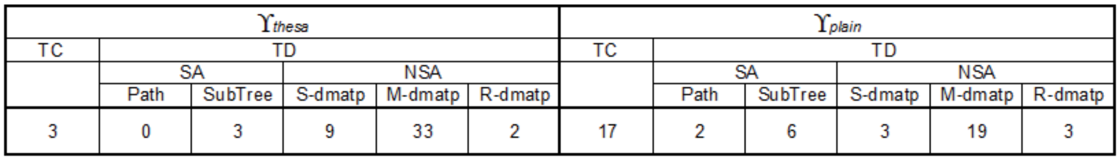
Topic extraction is performed through a customized and optimized proprietary version of the LDA model [4]. The Gibbs sampling procedure has been invoked with the following parameter values: $K = 50$, $\alpha = 1$ and $\beta = 0.01$; the number of sampling iterations equals 400.

The Gibbs sampling procedure has been run 10 times with different random initializations. The topics extracted through the last 9 runs were re-ordered to correspond as best as possible with the topics obtained through the first run. Correspondence was measured by the sum of the symmetrized Kullback Liebler (s-KL) distances.

The word list, associated with each extracted topic $T_e(j)$, consists of the M most frequent words so that the cumulate conditional probability is less than or equal to 0.2, i.e. $\sum_{j=1}^{M} P\left(w_i|z_i = j\right) \leq 0.2$ while including one more word brings to $\sum_{j=1}^{M+1} P\left(w_i|z_i = j\right) > 0.2$.

C. Automatic Labeling Of Topics

The ALOT algorithm is run with two different topics trees; namely a *plain topics tree* Υ_{plain} and a *thesaurized topics tree* Υ_{thesa}. The *plain topics tree* consists of $4,516$ topics, it is obtained from gDir by including nodes with a depth value less or equal than 5. Its topics are quite generic, while LDA extracted topics summarized through words lists consisting of technical words. Thus, the OpenOffice English Thesaurus



Υ_{thesa}						Υ_{plain}					
TC	TD					TC	TD				
	SA		NSA				SA		NSA		
	Path	SubTree	S-dmatp	M-dmatp	R-dmatp		Path	SubTree	S-dmatp	M-dmatp	R-dmatp
3	0	3	9	33	2	17	2	6	3	19	3

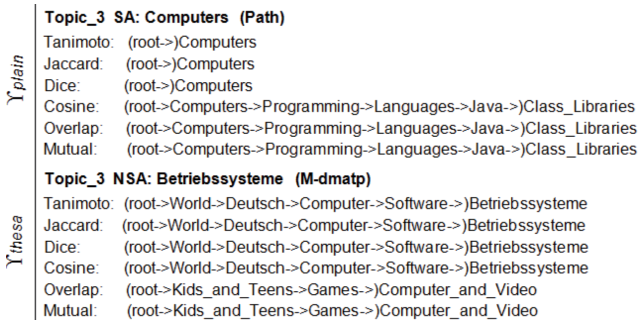
Fig. 3. ALOT results summary.

[19] has been exploited to obtain a *thesaurized topics tree* Υ_{thesa} built by using Wordnet [20] in such a way that for each word all its *senses* are recovered together with the following fields: generic terms (HYPERNYM), similar terms (SIMILAR), related terms (ALSO_SEE, PERTAINYM) and the antonym terms (ANTONYM). It is worthwhile to mention that in this paper Υ_{plain} and Υ_{thesa} are assumed to be fixed, i.e. they cannot be modified during the extraction phase. However, it is also possible to deal with the case where the topics tree can be extended, during the extraction phase, to include new knowledge.

In Figure 3, the number of times each ALOT's rule is used, to label the 50 LDA topics, is reported. The **NSA** labeling rules have been activated more often than **SA** labeling rules, for both topics trees. A possible explanation is as follows: the gDir Υ_{plain} tree has small depth and wide breadth. Therefore, it is difficult to find a unique predecessor for the same subtree (both as Subtree and as Path).

The best results have been achieved when excluding synonyms. In fact, while **SA** labeling rules are used more often than **NSA** labeling rules, the **TC** labeling rule has been used 17 times, more than one third of all the cases. A possible explanation for this is as follows: in the case where the word list of a topic is not semantically associated with the same context, the use of synonyms introduces distortion. For each word approximately 20 synonyms are used, thus resulting in a possible semantic divergence. Finally, the **R-dmatp** labeling rule associated with the failure of ALOT to provide a meaningful label is activated a few times, for both topics trees.

For matters of brevity, the labeling for only one topic is presented, i.e., Topic 3 whose 15 most frequent words are reported in Table I. Its words list consists of 47 words.



Υplain
Υthesa

Topic_3 SA: Computers (Path)
Tanimoto: (root->)Computers
Jaccard: (root->)Computers
Dice: (root->)Computers
Cosine: (root->Computers->Programming->Languages->Java->)Class_Libraries
Overlap: (root->Computers->Programming->Languages->Java->)Class_Libraries
Mutual: (root->Computers->Programming->Languages->Java->)Class_Libraries

Topic_3 NSA: Betriebssysteme (M-dmatp)
Tanimoto: (root->World->Deutsch->Computer->Software->)Betriebssysteme
Jaccard: (root->World->Deutsch->Computer->Software->)Betriebssysteme
Dice: (root->World->Deutsch->Computer->Software->)Betriebssysteme
Cosine: (root->World->Deutsch->Computer->Software->)Betriebssysteme
Overlap: (root->Kids_and_Teens->Games->)Computer_and_Video
Mutual: (root->Kids_and_Teens->Games->)Computer_and_Video

The label assigned to Topic 3 by using the *plain topics tree* Υ_{plain} is coherent with the one returned by ALOT when using the *thesaurized topics tree* Υ_{thesa}.

It is worthwhile to notice that the *thesaurized topics tree* Υ_{thesa} allows ALOT to establish a link between the term *Operative System* and the word *Betriebssysteme*, its meaning in German.This is probably due to the highly specific nature of the *thesaurized topics tree* Υ_{thesa} associated with *Operative System*, which results in a rich word list of synonyms capturing this context. It is worthwhile to mention that in many cases we found the equivalence of similarity measures with respect to ranking as reported by Omhover et al. [21].

V. Conclusions and Future Work

This paper describes an algorithm for automatic topic labeling according to a given topic hierarchy. It can be applied to any topic hierarchy summarized through a tree. The results of numerical experiments suggest that ALOT is effective. To the best of the authors' knowledge this is the first algorithm which maps extracted topic to topics labels associated with a topics hierarchy. Research directions include the development of new labeling rules, and the investigation of how performance is influenced by ALOT parameters.

Acknowledgment

The authors are grateful to the anonymous referees whose insightful comments helped us make significant improvements to the paper. Davide Magatti would like to thank DocFlow S.p.A. for funding his Ph.D. program.

TABLE I
Topic to be labeled.

Topic 3	0.0216
server	0.0197
microsoft	0.0096
linux	0.0069
domain	0.0068
metadata	0.0067
servers	0.0061
browser	0.0056
portal	0.0054
multimedia	0.0053
module	0.0046
developers	0.0046
java	0.0046
password	0.0045
download	0.0043
flash	0.0042

REFERENCES

[1] T. Hofmann, "Unsupervised learning by probabilistic latent semantic analysis," *Machine Learning*, vol. 42, no. 1-2, pp. 177–196, 2001.

[2] D. M. Blei, N. Andrew, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, January 2003.

[3] T. L. Griffiths and M. Steyvers, *Probabilistic Topic Models*, S. D. . W. K. T. Landauer, D. McNamara, Ed. Erlbaum, 2007.

[4] ——, "Finding scientific topics." *Proc Natl Acad Sci U S A*, vol. 101 Suppl 1, pp. 5228–5235, April 2004.

[5] M. Steyvers, P. Smyth, M. Rosen-Zvi, and T. Griffiths, "Probabilistic author-topic models for information discovery," in *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. New York, NY, USA: ACM Press, 2004, pp. 306–315.

[6] Q. Mei, C. Liu, H. Su, and C. Zhai, "A probabilistic approach to spatiotemporal theme pattern mining on weblogs," in *WWW '06: Proceedings of the 15th international conference on World Wide Web*. New York, NY, USA: ACM Press, 2006, pp. 533–542.

[7] T. Hofmann, "Probabilistic latent semantic analysis," in *In Proc. of Uncertainty in Artificial Intelligence, UAI99*, 1999, pp. 289–296.

[8] X. Wang and A. McCallum, "Topics over time: A non-markov continuous-time model of topical trends," in *Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2006, pp. 424–433.

[9] Q. Mei, X. Shen, and C. Zhai, "Automatic labeling of multinomial topic models," in *International Conference on Machine Learning (ICML)*, 2007.

[10] http://www.nlm.nih.gov/mesh/trees2008.html, October 2008.

[11] M. Bundschus, M. Dejori, S. Yu, V. Tresp, and H. P. Kriegel, "Statistical modeling of medical indexing processes for biomedical knowledge information discovery from text." in *Proceedings of the 8th International Workshop on Data Mining in Bioinformatics (BIOKDD '08)*, 2008.

[12] http://directory.google.com/, October 2008.

[13] http://www.criminal-law-lawyer-source.com/terms/accessory.html, April 2009.

[14] P. Cimiano and S. Staab, "Learning concept hierarchies from text with a guided hierarchical clustering algorithm," in *Proceedings of the ICML 2005 Workshop on Learning and Extending Lexical Ontologies with Machine Learning Methods*, C. Biemann and G. Paas, Eds., 2005.

[15] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[16] http://www.dmoz.org/, October 2008.

[17] R. Mizoguchi, "Tutorial on ontological engineering: part 3: Advanced course of ontological engineering," *New Gen. Comput.*, vol. 22, no. 2, pp. 198–220, 2004.

[18] S. Guadarrama and M. Garrido, "Concept-analizer: A tool for analyzing fuzzy concepts," in *Proceedings of IPMU'08*, M. O.-A. L. Magdalena and J. Verdegay, Eds., 2008, pp. 1084–1089.

[19] OpenOffice.org, http://www.openoffice.org/, 2008.

[20] C. Fellbaum, Ed., *WordNet: An Electronic Lexical Database (Language, Speech, and Communication)*. The MIT Press, May 1998.

[21] J. Omhover, M. Detyniecki, M. Rifqi, and B. Bouchon-Meunier, "Ranking invariance between fuzzy similarity measures applied to image retrieval," in *2004 IEEE International Conference on Fuzzy Systems, 2004. Proceedings*, vol. 3, 2004.