Towards Structure-Sensitive Hypertext Categorization

Alexander Mehler[1], Rüdiger Gleim[1], and Matthias Dehmer[2]

[1] Universität Bielefeld, 33615 Bielefeld, Germany
[2] Technische Universität Darmstadt, 64289 Darmstadt, Germany

Abstract. Hypertext categorization is the task of automatically assigning category labels to hypertext units. Comparable to text categorization it stays in the area of function learning based on the bag-of-features approach. This scenario faces the problem of a many-to-many relation between websites and their hidden logical document structure. The paper argues that this relation is a prevalent characteristic which interferes any effort of applying the classical apparatus of categorization to web genres. This is confirmed by a threefold experiment in hypertext categorization. In order to outline a solution to this problem, the paper sketches an alternative method of unsupervised learning which aims at bridging the gap between statistical and structural pattern recognition (Bunke et al. 2001) in the area of web mining.

1 Introduction

Web structure mining deals with exploring hypertextual patterns (Kosala and Blockeel (2000)). It includes the categorization of *macro structures* (Amitay et al. (2003)) such as web hierarchies, directories, corporate sites. It also includes the categorization of single web pages (Kleinberg (1999)) and the identification of page segments as a kind of structure mining on the level of *micro structures* (Mizuuchi and Tajima (1999)). The basic idea is to perform structure mining as function learning in order to map web units *above*, *on* or *below* the level of single pages onto at most one predefined category per unit (Chakrabarti et al. (1998)). The majority of these approaches utilizes text categorization methods. But other than text categorization, they also use HTML markup, metatags and link structure beyond bag-of-word representations of the pages' wording as input of feature selection (Yang et al. (2002)). Chakrabarti et al. (1998) and Fürnkranz (2002) extend this approach by including pages into feature selection which are interlinked with the page to be categorized. Finally, the aggregation of representations of the wording, markup and linking of pages is demonstrated by (Joachims et al. (2001)).

The basic assumption behind these approaches is as follows: Web units of similar function/content tend to have similar structures. The central problem is that these structures are not directly accessible by segmenting and categorizing *single web pages*. This is due to *polymorphism* and its reversal relation of *discontinuous manifestation* (Mehler et al. (2004)): Generally speaking, polymorphism occurs if the same (hyper-)textual unit manifests

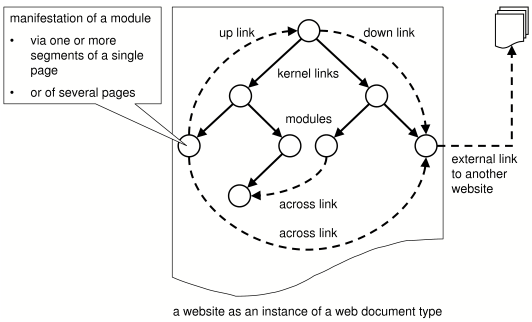

Fig. 1. Types of links connecting hypertext modules (symbolized as circles).

several categories. This one-to-many relation of expression and content units is accompanied by a reversal relation according to which the same content or function unit is distributed over several expression units. This combines to a many-to-many relation between explicit, manifesting web structure and implicit, manifested functional or content-based structure.

Polymorphism occurs when, for example, the same web page of the genre of conference websites provides information about the call for papers, the submission procedure and conference registration, that is, when it manifests more than one function. The reversal case occurs when, for example, a call for papers is manifested by different pages each informing about another conference topic. The former case results in multiple categorizations without corresponding to ambiguity of category assignment since actually several categories are manifested. The latter case results in defective or even missing categorizations since the pages manifest features of the focal category only in part. Both cases occur simultaneously, if a page manifests several categories, but some of them only in part. If this many-to-many relation is prevalent, proper hypertext categorization is bound to a preliminary structure analysis which first resolves polymorphism and discontinuous manifestation. We hypothesize this structure analysis to be constrained as follows:

- The functional structure of websites is determined by their membership in *web genres* (Yoshioka and Herman 2000). Hypertext categories are (insofar as they focus on the functions web pages are intended to serve) specific to genres, e.g. *conference website* (Yoshioka and Herman 2000), *personal home page* (Rehm (2002)) or *online shop*.
- What is common to instances of different web genres is the existence of an implicit *logical document structure* (LDS) – analogously to textual units whose LDS is described in terms of section, paragraph and sentence categories. In case of instances of web genres we hypothesize that their LDS includes at least three levels:
  - Document types, which are typically manifested by websites, constitute the level of pragmatically closed acts of web-based communica-

tion (e.g. conference organization or online shopping). They organize a system of dependent sub-functions manifested by modules:
  - Module types are functionally homogeneous units of web-based communication manifesting a single, but dependent function, e.g. *call for papers*, *program* or *conference venue* as sub-functions of the function of *web-based conference organization*.
  - Finally, elementary building blocks (e.g. logical *lists*, *tables*, *sections*) only occur as dependent parts of hypertext modules.
- Uncovering the LDS of websites contributes to breaking the many-to-many-relation of polymorphism and discontinuous manifestation. It aims to explicate which modules are manifested by which (segments of which) visible web pages of the same site and which links of which types – as distinguished in figure (1) – interlink these modules.
- The central hypothesis of this paper is that hypertext categorization has to be reconstructed as a kind of *structure learning* focussing on prototypical, recurrent patterns of the LDS of websites as instances of web genres on the level of document types and their typing according to the functions their constitutive modules are intended to serve.

In order to support this argumentation, the following section describes an experiment in hypertext categorization. After that, an algorithm is outlined which reconstructs hypertext categorization in terms of a structure sensitive model. Finally, we give some conclusions and prospect future work.

2 An Experiment in Hypertext Categorization

Our hypothesis is that if polymorphism is a prevalent characteristic of web units, web pages cannot serve as input of categorization since polymorphic pages simultaneously instantiate several categories. Moreover, these multiple categorizations are not simply resolved by segmenting the focal pages, since they possibly manifest categories only discontinuously so that their features do not provide a sufficient discriminatory power. In other words: We expect polymorphism and discontinuous manifestation to be accompanied by many multiple categorizations without being reducible to the problem of disambiguating category assignments. In order to show this, we perform a categorization experiment according to the classical setting of function learning, using a corpus of the genre of *conference websites*. Since these websites serve recurrent functions (e.g. *paper submission*, *registration* etc.) they are expected to be structured homogeneously on the basis of stable, recurrent patterns. Thus, they can be seen as good candidates of categorization.

The experiment is performed as follows: We apply support vector machine (SVM) classification which proves to be successful in case of sparse, high dimensional and noisy feature vectors (Joachims (2002)). SVM classification is performed with the help of the LibSVM (Hsu et al. (2003)).

Category	rec.	prec.
Abstract(s)	0.2	1.0
Accepted Papers	0.3	1.0
Call for Papers	0.1	1.0
Committees	0.5	0.8
Contact Information	0	0
Exhibition	0.4	1.0
Important Dates	0.8	1.0
Invited Talks	0	0

Category	rec.	prec.
Menu	0.7	0.7
Photo Gallery	0	0
Program, Schedule	0.8	1.0
Registration	0.9	1.0
Sections, Sessions, Plenary etc.	0.1	0.3
Sponsors and Partners	0	0
Submission Guidelines etc.	0.5	0.8
Venue, Travel, Accommodation	0.9	1.0

Table 1. The categories of the *conference website genre* applied in the experiment.

We use a corpus of 1,078 English conference websites (for the list of URLs see `http://ariadne.coli.uni-bielefeld.de/indogram/`) and 28,801 web pages. Hypertext representation is done by means of a bag-of-features approach using about 85,000 lexical and 200 HTML features. This representation was done with the help of the HyGraph system (Gleim (2005)) which automatically explores websites and maps them onto hypertext graphs (Mehler et al. (2004)). Following Hsu et al. (2003), we use a R*adial* B*asis* F*unction* kernel instead of a polynomial kernel, but other than in Mehler et al. (2004), we augment the corpus base and use a more fine-grained category set. Optimal parameter selection is based on a minimization of a 5-fold cross validation error. Furthermore, we perform a binary categorization for each of the 16 categories based on 16 training sets of positive/negative examples – see table (1). The overall size of the training set is 1,858 pages (of 284 websites); the size of the test set is 200 (of 82 websites). We perform three experiments:

Experiment A – one against all: First we apply a one against all strategy, that is, we use $X \setminus Y_i$ as the set of negative examples for learning category C_i where X is the set of all training examples and Y_i is the set of positive examples of C_i. The results are listed in table (1). It shows the expected low level of effectivity: recall and precession perform very low on average. In three cases the classifiers fail completely. This result is confirmed when looking at column A of table (2): It shows the number of pages with up to 7 category assignments. In the majority of cases no category could be applied at all – only one-third of the pages was categorized.

Experiment B – lowering the discriminatory power: In order to augment the number of categorizations, we lowered the categories' selectivity by restricting the number of negative examples per category to the number of the corresponding positive examples by sampling the negative examples according to the sizes of the training sets of the remaining categories. The results are shown in table (2): The number of zero categorizations is dramatically reduced, but at the same time the number of pages mapped onto more than one category increases dramatically. There are even more than 1,000 pages which are mapped onto more than 5 categories.

Experiment C – segment level categorization: Thirdly, we apply the classifiers trained on the monomorphic training pages on segments derived as

#categorizations	A (page level)	B (page level)	C (segment level)
0	12,403	346	27,148
1	6,368	2387	9,354
2	160	5076	137
3	6	5258	1
4	0	3417	0
5	0	923	0
6	0	1346	0
7	0	184	0

Table 2. The number of pages mapped onto $0, 1, ..., 7$ categories in exp. A,B,C.

follows: Pages are segmented into spans of at least 30 tokens reflecting segment borders according to the third level of the pages' document object model (DOM) trees. Column C of table (2) shows that this scenario does not solve the problem of multiple categorizations since it falls back into the problem of zero categorizations. In other words: polymorphism is not resolved by simply segmenting pages, as alternative segmentations along the same line of constraints have confirmed.

There are several competing interpretations of these results: The category set may be judged to be wrong, but actually, it reflects the most differentiated set of categories applied so far in this area (cf. Yoshioka and Herman (2000)). Secondly, the representation model may be judged to be wrong, but actually it is usually applied in (hyper-)text categorization. Thirdly, the categorization method may be seen to be ineffective, but SVM classification is known to be one of the most effective methods in the area under consideration. Further, the classifiers may be judged to be wrong – of course the training set could be enlarged, but it already includes about 2,000 monomorphic training units. Finally, the focal units (i.e. web pages) may be judged to be unsystematically polymorph in the sense that they manifest several logical units or the same logical, functionally homogeneous unit is distributed over several web pages. It is this last interpretation which we believe to be supported by the categorization experiment.

Why are linear segmentations of web pages according to experiment C insufficient? The reason is twofold: Because a category may be distributed over several pages, it is possible that pages analyzed in isolation do not manifest category markers sufficiently. Thus, segmentations of pages of the same site are interdependent. But since not all pages belong to the same structural level of a website (a call for participation belongs to another level than an abstract of one of the invited talks), segmentation also needs to be aware of website structure. As categories are manifested by pages of different structural levels, these pages are not linearly ordered. This is also proven by structural recursion, since a call for papers, for example, may include several workshops each having its own call for papers. That is, linear segmentations of pages do not suffice because of discontinuous manifestations. But linear orderings of pages do not suffice either because of functional website structure. Fuzzy classification does not offer immediately a solution to this problem as long as

it only performs multiple category assignments to varying degrees of membership, since such mappings do not allow distinguishing between ambiguity of category assignment and polymorphism (which is not to be confused with ambiguity). Thus, web page categorization has to include the resolution of polymorphism and discontinuous manifestation and thus relates to learning *implicit logical hypertext document structure* (LDS). The following section outlines an algorithm for the induction of this LDS *in conceptual terms*.

3 Reconstructing Hypertext Categorization

A central implication of latter section is that, prior to hypertext categorization, the many-to-many relation of visible and hidden web structure has to be resolved at least with respect to LDS. Thus, hypertext categorization is bound to a structural analysis. Insofar this analysis results in structured representations of web units, function learning as performed by text categorization is inappropriate to mining web genres. It unsystematically leads to multiple categorizations when directly applied to web units whose borders do not correlate with the functional or content-based categories under consideration. Rather, a sort of structure learning has to be performed, mapping these units onto representations of their LDS which only then are object to mining prototypical sub-structures of web genres. In this section, hypertext categorization is reconstructed along this line of argumentation. An algorithm is outlined, henceforth called *LDS algorithm*, which addresses structure learning from the point of view of prototypes of LDS. It is divided into two parts:

I. Logical Document Structure Learning Websites as supposed instances of web genres have first to be mapped onto representations of their LDS. That is polymorphism has to be resolved with respect to constituents of this structure level. This includes the following tasks:

- Visible segments of web pages have to be identified as manifestations of constituents of LDS.
- Visible hyperlinks have to be identified as manifestations of logical links, i.e. as kernel links or up, down, across or external links.
- Finally, functional equivalents of hyperlinks have to be identified as manifestations of logical links according to the same rules, i.e. of links without being manifested by hyperlinks.

Solving these tasks, each website is mapped onto a representation of its LDS based on the building blocks described in section (1). This means that websites whose visible surface structures differ dramatically may nevertheless be mapped onto similar LDS representations, and vice versa. So far, these intermediary representations lack any typing of their nodes, links and sub-structures in terms of functional categories of the focal web genre. This functional typing is addressed by the second part of the algorithm:

II. Functional Structure Learning The representations of LDS are input to an algorithm of *computing with graphs* which includes four steps:

1. *Input:* As input we use a corpus $C = \{G_i \mid i \in I\}$ of labeled *typed directed graphs* $G_i = (V, E, k(G_i), \tau)$ with kernel hierarchical structure modeled by an ordered rooted tree $k(G_i) = (V, D, x, \mathcal{O})$ with root x and order relation $\mathcal{O} \subseteq D^2$, $D \subseteq E$. $\mathcal{O}$ is an ordering of kernel links $e \in D$ only. Since $k(G_i)$ is a rooted tree, it could equivalently be defined over the nodes. Typing of edges $e \in E$ is done by a function $\tau : E \to T$ where T is a set of type labels. In case of websites, vertices $v \in V$ are labeled as either accessible or unaccessible web pages or resources and edges are typed as kernel, across, up, down, internal, external or broken links. In case of logical hypertext document structure, vertices are logical modules whereas the set of labels of edge types remains the same.
2. *Graph similarity measuring:* The corpus C of graphs is input to a similarity measure $s\colon C^2 \to [0, 1]$ used to built a similarity matrix (Bock (1974)) $\boldsymbol{S} = (s_{kj})$ where s_{kj} is the similarity score of the pairing $G_i, G_j \in C$. s has to be sensitive to the graphs' kernel hierarchical structure as well as to the labels of their vertices and the types of their edges. We utilize the measure of Dehmer et al. (2004) which is of cubic complexity.
3. *Graph clustering:* Next, the similarity matrix is input to clustering, that is, to unsupervised learning without presetting the number of classes or categories to be learned. More specifically, we utilize hierarchical agglomerative clustering (Bock (1974)) based on average linkage with subsequent partitioning. This partitioning refers to a lower bound (Rieger (1989)) $\theta = \bar{\eta} + \frac{1}{2}\sigma$, where $\bar{\eta}$ is the mean and σ the standard deviation of the absolute value of the differences of the similarity levels of consecutive agglomeration steps. This gives a threshold for selecting an agglomeration step for dendrogram partitioning whose similarity distance to the preceding step is greater than θ. We use the first step exceeding θ.
4. *Graph prototyping:* Next, for each cluster $X = \{G_{i_1}, \ldots, G_{i_n}\} \subseteq C$ of the output partitioning of step (3) a graph median $\hat{G}$ has to be computed according to the approach of Bunke et al. (2001):

$$\hat{G} = \arg\max_{G \in X} \frac{1}{n} \sum_{k}^{|X|} s(G, G_{i_k}) \tag{1}$$

   The basic idea of applying this formula is to use $\hat{G}$ as a prototype of the cluster X in the sense that it prototypically represents the structuring of all members of that set of graphs.
5. *Graph extraction:* The last step is to use the prototypes $\hat{G}$ as kernels of instance based learning. More specifically, the prototype graphs can be used as templates to extract sub-structures in new input graphs. The idea is to identify inside these graphs recurrent patterns and thus candidates of functional categories of the focal genre (e.g. *paper submission* or *conference venue* graphs in case of the genre of *conference websites*).

It is this last step which addresses the final categorization by using *structured categories in order to categorize sub-structures of the input graphs.* It replaces the mapping of visible segments of web units onto predefined categories by mapping sub-structures of the hidden LDS onto clusters of homogeneously structured instances of certain module types of the focal web genre.

4 Structure-Based Categorization

This section presents a preliminary evaluation of graph similarity measuring as part of the LDS algorithm of section (3). Since this algorithm was presented as a *conceptual* prospect on how to solve the problem whose relevance was *empirically* proven in section (2), we focus on an experiment solely based on the D*ocument* O*bject* M*odel* (DOM) trees of monomorphic web pages completely ignoring their text content. That is we concentrate on part II of the algorithm presupposing part I to be already solved. The aim is to examine how far we get when categorizing web pages along functional categories by looking for their document structure only. This is all but not trivial. A positive evaluation can be seen to motivate an implementation of the LDS algorithm which includes graph similarity measuring as its kernel.

In our experiment, we chose a corpus of wikipedia articles addressing the following categories: *American Presidents* (41 pages), *European Countries* (50), *German Cities* (78) and *German Universities* (93). We restrict the representation of these articles to their DOM trees excluding all text nodes (i.e. only HTML-tags are taken into account). Then we use the similarity measure of Dehmer et al. (2004), perform pairwise comparisons of the DOM trees and derive a corresponding distance matrix. Next, we apply two clustering methods: hierarchical agglomerative and k-means clustering. Because of knowing the category of each article we can perform an exhaustive parameter study to maximize the well known efficiency measures *purity*, *inverse purity* and the combined F*-measure*. Hierarchical agglomerative clustering does not need any information on the expected number of clusters so we examined all possible clusterings and chose the one maximizing the F-measure. In contrast to this, k-means needs to be informed about the number of clusters in advance, which in the present experiment equals 4.

Table (3) summarizes our findings. The general picture is that hierarchical clustering with subsequent partitioning leads to better results than k-means. The best F-measure (i.e. 0.643) was realized by a partitioning into 7 clusters. The partitioning with exactly four clusters performs only a little worse (i.e. 0.611). The best k-means clustering yields an F-measure 0.601. In order to provide a baseline for better rating these results, we performed a random clustering. This leads to an F-measure of 0.311 (averaged over 1,000 runs). Content-based categorization experiments using the bag of words model have reported F-measures of about 0.86 (Yang (1999)). We conclude that analyzing document structure provides a remarkable amount of infor-

Clustering Algorithm	# Clusters	F-Measure	Purity	Inverse Purity	Method-Specifical
hierarchical	7	0.643	0.718	0.553	average linkage
hierarchical	4	0.611	0.607	0.718	complete linkage
k-means	4	0.601	0.595	0.592	squared Euclidean distance
random	4	0.311	0.362	0.312	

Table 3. Evaluation results.

mation to categorization which—in combination with textual content of web pages—should be verified as a candidate for replacing or at least supporting the traditional SVM setting. As this nevertheless presupposes a resolution of polymorphism and discontinuous manifestation, it can only be seen as a preliminary step of implementing the LDS algorithm which we propose as a candidate to overcome the presented problems of hypertext categorization.

5 Conclusion

This paper argued that the structure of websites is an uncertain manifestation of their hidden logical document structure. As far as hypertext categorization deals with functional, genre-based categories, the visible structure does not suffice as input to categorization because of its many-to-many relation to the LDS. Thus, a prerequisite of hypertext categorization is a reconstruction of this LDS. A categorization experiment has been performed in order to indirectly show the impact of polymorphism and discontinuous manifestations. In order to find a solution to this problem, hypertext categorization has been conceptually reconstructed by means of an algorithm which reflects the difference of visible and hidden structure and utilizes the paradigm of structure instead of function learning. Future work aims at implementing this algorithm.

Bibliography

AMITAY, E. and CARMEL, D. and DARLOW, A. and LEMPEL, R. and SOFFER, A. (2003): The connectivity sonar. *Proc. of the 14th ACM Conference on Hypertext, 28−47.*

BOCK, H.H. (1974): *Automatische Klassifikation.* Vandenhoeck & Ruprecht, Göttingen.

BUNKE, H. and GÜNTER, S. and JIANG, X. (2001): Towards bridging the gap between statistical and structural pattern recognition. *Proc. of the 2nd Int. Conf. on Advances in Pattern Recognition, Berlin, Springer, 1−11.*

CHAKRABARTI, S. and DOM, B. and INDYK, P. (1998): Enhanced hypertext categorization using hyperlinks. *Proc. of ACM SIGMOD, International Conf. on Management of Data, ACM Press, 307−318.*

DEHMER, M. and GLEIM, R. and MEHLER, A. (2004): A new method of similarity measuring for a specific class of directed graphs. *Submitted to Tatra Mountain Journal, Slovakia.*

FÜRNKRANZ, J. (2002): Hyperlink ensembles: a case study in hypertext classification. *Information Fusion, 3(4), 299−312.*

GIBSON, D. and KLEINBERG, J. and RAGHAVAN, P. (1998): Inferring web communities from link topology. *Proc. of the 9th ACM Conf. on Hypertext, 225−234.*

GLEIM, R. (2005): Ein Framework zur Extraktion, Repräsentation und Analyse webbasierter Hypertexte, *Proc. of GLDV '05, 42−53.*

HSU, C.-W. and CHANG, C.-C. and LIN, C.-J. (2003): A practical guide to SVM classification. *Technical report, Department of Computer Science and Information Technology, National Taiwan University.*

JOACHIMS, T. (2002): *Learning to classify text using support vector machines.* Kluwer, Boston, 2002.

JOACHIMS, T. and CRISTIANINI, N. and SHAWE-TAYLOR, J. (2001): Composite kernels for hypertext categorisation. *Proc. of the 11th ICML, 250−257.*

KLEINBERG, J. (1999): Authoritative sources in a hyperlinked environment. *Journal of the ACM, 46(5), 604−632*

KOSALA, R. and BLOCKEEL, H. (2000): Web mining research: A survey. *SIGKDD Explorations, 2(1), 1−15.*

MEHLER, A. and DEHMER, M. and GLEIM, R. (2004): Towards logical hypertext structure − a graph-theoretic perspective. *Proc. of I2CS '04, Berlin, Springer.*

MIZUUCHI, Y. and TAJIMA, K. (1999): Finding context paths for web pages. *Proc. of the 10th ACM Conference on Hypertext and Hypermedia, 13−22.*

REHM, G. (2002): Towards automatic web genre identification. *Proc. of the Hawai'i Int. Conf. on System Sciences.*

RIEGER, B. (1989): *Unscharfe Semantik.* Peter Lang, Frankfurt a.M.

YANG, Y. (1999): An Evaluation of Statistical Approaches to Text Categorization. *Journal of Information Retrieval, 1, 1/2*, 67−88.

YANG, Y. and SLATTERY, S. and GHANI, R. (2002): A study of approaches to hypertext categorization. *Journal of Intelligent Information Systems, 18(2-3), 219−241.*

YOSHIOKA, T. and HERMAN, G. (2000): Coordinating information using genres. *Technical report, Massachusetts Institute of Technology.*