Web Corpus Mining by instance of Wikipedia

Rüdiger Gleim, Alexander Mehler & Matthias Dehmer
Bielefeld University, D-33615 Bielefeld, Germany
Ruediger.Gleim@uni-bielefeld.de
Alexander.Mehler@uni-bielefeld.de
Technische Universität Darmstadt, Fachbereich Informatik
dehmer@tk.informatik.tu-darmstadt.de

Abstract

In this paper we present an approach to structure learning in the area of web documents. This is done in order to approach the goal of webgenre tagging in the area of web corpus linguistics. A central outcome of the paper is that purely structure oriented approaches to web document classification provide an information gain which may be utilized in combined approaches of web content and structure analysis.

1 Introduction

In order to reliably judge the collocative affinity of linguistic items, it has to be considered that judgements of this kind depend on the scope of certain genres or registers. According to Stubbs (2001), words may have different collocates in different *text types* or *genres* and therefore may signal one of those genres when being observed. Consequently, corpus analysis requires, amongst others, a comparison of occurrences in a given text with typical occurrences in other texts of the same genre (Stubbs, 2001, p. 120).

This raises the question how to judge the membership of texts, in which occurrences of linguistic items are observed, to the genres involved. Evidently, because of the size of the corpora involved, this question is only adequately answered by reference to the area of *automatic* classification. This holds all the more for web corpus linguistics (Kilgarriff and Grefenstette, 2003; Baroni and Bernardini, 2006) where large corpora of web pages, whose membership in *webgenres* is presently unknown, have to be analyzed. Consequently, web corpus linguistics faces two related task:

1. *Exploration:* The task of initially *exploring* which webgenres actually exist.
2. *Categorization:* The task of *categorizing* hypertextual units according to their membership in the genres being explored in the latter step.

In summary, web corpus linguistics is in need of webgenre-sensitive corpora, that is, of corpora in which for the textual units being incorporated the membership to webgenres is annotated. This in turn presupposes that these webgenres are first of all explored.

Currently to major classes of approaches can be distinguished: On the one hand, we find approaches to the categorization of *macro structures* (Amitay et al., 2003) such as web hierarchies, directories and corporate sites. On the other hand, this concerns the categorization of *micro structures* as, for example, single web pages (Kleinberg, 1999) or even page segments (Mizuuchi and Tajima, 1999). The basic idea of all these approaches is to perform categorization as a kind of function learning for mapping web units *above*, *on* or *below* the level of single pages onto at most one predefined category (e.g. genre label) per unit (Chakrabarti et al., 1998). Thus, these approaches focus on the categorization task while disregarding the exploration task. More specifically, the majority of these approaches utilizes *text categorization* methods in conjunction with HTML markup, metatags and link structure beyond bag-of-word representations of the pages' wording as input of feature selection (Yang et al., 2002) – in some cases also of linked pages (Fürnkranz, 1999).

What these approaches are missing is a more general account of web document structure as a source of genre-oriented categorization. That is, they solely map web units onto feature vectors by disregarding their structure. This includes linkage beyond pairwise linking as well as document internal structures according to the Document Object Model (DOM). A central pitfall of this approach is that it disregards the impact of genre membership to document structure and, thus, the signalling of the former by the latter (Ventola, 1987). Therefore a structure-sensitive approach is needed in the area of corpus linguistics which allows for automatic webgenre tagging. That is, an approach which takes both levels of structuring of web documents into account: On the level of their hyperlink-based linkage *and* on the level of their internal structure.

In this paper we present an algorithm as a preliminary step for tackling the exploration and categorization task together. More specifically, we present an approach to unsupervised structure learning which uses tree alignment algorithms as similarity kernels and cluster analysis for class detection. The paper includes a comparative study of several approaches to tree alignment as a source of similarity measuring of web documents. Its central topics are:

- *To what extent is it possible to predict the membership of a web document in a certain genre (or register) solely on grounds of its structure when its lexical content and other content bearing units are completely deleted?* In other words, we ask to what extent structure signals membership in genre.

- A more methodical question regards the choice of appropriate measures of structural similarity to be included into structure learning. In this context, we comparatively study several variants of measuring similarities of trees, that is, *tree edit distance* as well as a class of algorithms which are based on tree linearizations as input to *sequence alignment*.

Our overall findings hint at two critical points: First, there is a significant contribution of structure-oriented methods to webgenre categorization which is unexplored in predominant approaches. Second, and most surprisingly, all methods analyzed toughly compete with a method based on random linearization of input documents.

Why is this research important for web corpus linguistics? An answer to this question can be outlined as follows:

- We explore a further resource of reliably tagging web genres and registers, respectively, in the form of document structure.

- We further develop the notion of *webgenre* and thus help to make document structure accessible to collocation and other corpus linguistic analyses.

In order to support this argumentation, we first present a structure insensitive approach to web categorization in section (2). It shows that this insensitivity systematically leads to multiple categorizations which cannot be traced back to ambiguity of category assignment. In order to solve this problem, an alternative approach to structure learning is presented in sections (3.1), (3.2) and (3.3). This approach is evaluated in section (3.4) on grounds of a corpus of Wikipedia articles. The reason for utilizing this test corpus is that the content-based categories which the explored web documents belong to are known so that we can apply the classical apparatus of evaluation of web mining. The final section concludes and prospects future work.

2 Hypertext Categorization

The basic assumption behind present day approaches to hypertext categorization 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*: Generally speaking, polymorphism occurs if the same (hyper-)textual unit manifests 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.

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). We use a corpus of 1,078 English conference websites 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 which explores websites and maps them onto hypertext graphs (Mehler and Gleim, 2005). Following (Hsu et al., 2003), we use a R*adial* B*asis* F*unction* kernel and make optimal parameter selection based on a minimization of a 5-fold cross validation error. Further, we perform a binary categorization for each of the 16 categories based on 16 training sets of pos./neg. examples (see table 1). The size of the training set is 1,858 pages (284 sites); the size of the test set is 200 (82 sites). We perform 3 experiments:

1. *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.

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
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.

2. *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.

3. *Experiment C – segment level categorization:* Thirdly, we apply the classifiers trained on the monomorphic training pages on segments derived as 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 trees. Column C of table (2) shows that this scenario does not solve the problem of multiple categorizations since it falls back to the problem of zero categorizations. Thus, polymorphism is not resolved by simply segmenting pages, as other segmentations along the same line of constraints confirmed.

There are competing interpretations of these results: The category set may be judged to be wrong. But it reflects the most differentiated set applied so far in this area. Next, the representation model

number of 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 experiment A, B and C.

may be judged to be wrong, but actually it is usually applied in text categorization. Third, the categorization method may be seen to be ineffective, but SVMs are known to be one of the most effective methods in this area. Further, the classifiers may be judged to be wrong – of course the training set could be enlarged, but already includes about 2,000 manually selected monomorphic training units. Finally, the focal units (i.e. web pages) may be judged to be unsystematically polymorph in the sense of manifesting several logical units. It is this interpretation which we believe to be supported by the experiment.

If this interpretation is true, the structure of web documents comes into focus. This raises the question, what can be gained at all when exploring the visible structuring of documents as found on the web. That is, what is the information gain when categorizing documents solely based on their structures. In order to approach this question we perform an experiment in structure-oriented classification in the next section. As we need to control the negative impact of polymorphism, we concentrate on monomorphic pages which uniquely belong to single categories. This can be guaranteed with the help of Wikipedia articles which, with the exception of special disambiguation pages, only address one topic respectively.

3 Structure-Based Categorization

3.1 Motivation

In this section we investigate how far a corpus of documents can be categorized by solely considering the explicit document structure without any textual content. It is obvious that we cannot expect the results to reach the performance of content based approaches. But if this approach allows to significantly distinguish between categories in contrast to a reference random decider we can conclude that the involvement of structure information may positively affect categorization performance. A positive evaluation can be seen to motivate an implementation of the *Logical Document Structure (LDS) algorithm* proposed by Mehler et al. (2005) who include graph similarity measuring as its kernel. We expect the same experiment to perform significantly better on the LDS instead of the explicit structures. However this experiment can only be seen as a first attempt. Further studies with larger corpora are required.

3.2 Experiment setup

In our experiment, we chose a corpus of articles from the German Wikipedia addressing the following categories:

- *American Presidents* (41 pages)
- *European Countries* (50 pages)
- *German Cities* (78 pages)
- *German Universities* (93 pages)

With the exception of the first category most articles, being represented as a HTML web page, share a typical, though not deterministic visible structure. For example a Wikipedia article about a city contains an info box to the upper right which lists some general information like district, population and geographic location. Furthermore an article about a city contains three or more sections which address the history, politics, economics and possibly famous buildings or persons. Likewise there exist certain design *guidelines* by the Wikipedia project to write articles about countries and universities. However these guidelines are not always followed or they are adapted from one case to another. Therefore, a categorization cannot rely on definite markers in the content. Nevertheless, the expectation is that a human reader, once he has seen a few samples of each category, can with high probability guess the category of an article by simple looking at the layout or visible structure and ignoring the written content. Since the layout (esp. the structure) of a web page is encoded in HTML we consider the structure of their DOM[1]-trees for our categorization experiment. If two articles of the same category share a common visible structure, this should lead to a significant similarity of

[1] D*ocument* O*bject* M*odel*.

the DOM-trees. The articles of category 'American Presidents' form an exception to this principle up to now because they do not have a typical structure. The articles about the first presidents are relatively short whereas the articles about the recent presidents are much more structured and complex. We include this category to test how well a structure based categorizer performs on such diverse structurations. We examine two corpus variants:

I. All HTML-Tags of a DOM-tree are used for similarity measurement.

II. Only those HTML-tags of a DOM-tree are used which have an impact on the visible structure (i.e. inline tags like *font* or *i* are ignored).

Both cases, I and II, do not include any text nodes. That is, all lexical content is ignored. By distinguishing these two variants we can examine what impact these different degrees of expressiveness have on the categorization performance.

3.3 Distance measurement and clustering

The next step of the experiment is marked by a pairwise similarity measurement of the wikipedia articles which are represented by their DOM-trees according to the two variants described in section 3.2. This allows to create a distance matrix which represents the (symmetric) distances of a given article to any other. In a subsequent and final step the distance matrix will be clustered and the results analyzed.

How to measure the similarity of two DOM-trees? This raises the question what *exactly* the subject of the measurement is and how it can be adequately modeled. Since the DOM is a tree and the order of the HTML-tags matters, we choose ordered trees. Furthermore we want to represent what tag a node represents. This leads to ordered labeled trees for representation. Since trees are a common structure in various areas such as image analysis, compiler optimization and bio informatics (i.e. RNA analysis) there is a high interest in methods to measure the similarity between trees (Tai, 1979; Zhang and Shasha, 1989; Klein, 1998; Chen, 2001; Höchsmann et al., 2003). One of the first approaches with a reasonable computational complexity was introduced by Tai (1979) who extended the problem of sequence edit distance to trees.

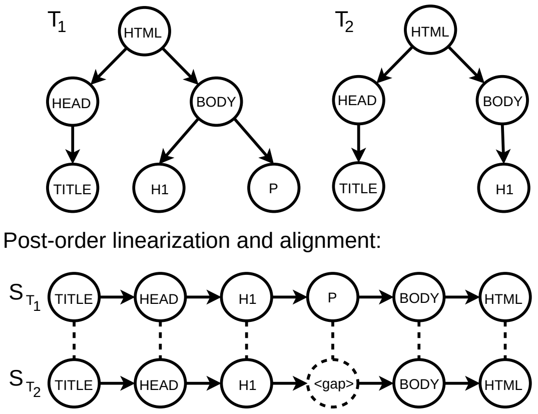

Figure 1: An example for Post-order linearization and sequence alignment.

The following description of tree edit distances is due to Bille (2003): The principle to compute the edit distance between two trees T_1, T_2 is to successively perform elementary edit operations on the former tree to turn it into the formation of the latter. The edit operations on a given tree T are as follows: *Relabel* changes the label of a node $v \in T$. *Delete* deletes a non-root node $v \in T$ with a parent node $w \in T$. Since v is being deleted, its child nodes (if any) are inserted as children of node w. Finally the *Insert* operation marks the complement of delete. Next, an *edit script* S is a list of consecutive edit operations which turn T_1 into T_2. Given a cost function for each edit operation the cost of S is the sum of its elementary operation costs. The *optimal edit script* (there is possibly more than one) between T_1 and T_2 is given by the edit script of minimum cost which equals the *tree edit distance*.

There are various algorithms known to compute the edit distance (Tai, 1979; Zhang and Shasha, 1989; Klein, 1998; Chen, 2001). They vary in computational complexity and whether they can be used for general purpose or under special restrictions only (which allows for better optimization). In this experiment we use the general-purpose algorithm of Zhang and Shasha (1989) which shows a complexity of $O(|T_1||T_2|min(L_1, D_1)min(L_2, D_2))$ where $|T_i|$, L_i, D_i denote the number of nodes, the number of leafs and the depth of the trees respectively.

The approach of tree edit distance forms a good balance between accurate distance measurement of trees and computational complexity. However, especially for large corpora it might be useful to examine how well other (i.e. faster) methods perform. We therefore consider another class of algorithms for distance measurement which are based on sequence alignments via dynamic programming. Since this approach is restricted to the comparison of sequences, a suitable linearization of the DOM trees has to be found. For this task we use several strategies of tree node traversal: Pre-Order, Post-Order and Breath-First-Search (BFS) traversal. Figure (1) shows a linearization of two sample trees using Post-Order and how the resulting sequences S_{T_i} may have been aligned for the best alignment distance. We have enhanced the labels of the linearized nodes by adding the in- and out degrees corresponding to the former position of the nodes in the tree. This information can be used during the computation of the alignment cost: An example of this is that the alignment of two nodes with identical HTML-tags but different in/out degrees will result in a higher cost than in cases where these degrees match. Following this strategy, at least part of the structure information is preserved. This approach is followed by Dehmer (2005) who develops a special form of tree linearization which is based on tree levels.

Obviously, a linearization poses a loss of structure information which has impact on the results of distance measurement. But the computational complexity of sequence alignments ($O(n^2)$) is significantly better than of tree edit distances. This leads to a trade-off between the expressiveness of the DOM-Tree representation (in our case tree vs. linearization to a sequence) and the complexity of the algorithms to compute the distance thereon. In order to have a baseline for tree linearization techniques we have also tested random linearizations. According to this method, trees are transformed into sequences of nodes in random order. For our experiment we have generated 16 random linearizations and computed the median of their categorization performances.

Next, we perform pairwise distance measurements of the DOM-trees using the set of algorithms described above. We then apply two clustering methods on the resulting distance matrices: hierarchical agglomerative and k-means clustering. Hierarchical agglomerative clustering does not need any information on the expected number of clusters so we examine all possible clusterings and chose the one maximizing the F-measure. However we also examine how well hierarchical clustering performs if the number of partitions is restricted to the number of categories. In contrast to the previous approach, k-means needs to be informed about the number of clusters in advance, which in the present experiment equals the number of categories, which in our case is four. Because we know 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*.

3.4 Results and discussion

The tables (3) and (5) show the results for corpus variant I (using all HTML-tags) and variant II (using structure relevant HTML-tags only) (see section 3.2). The general picture is that hierarchical clustering performs significantly better than k-means. However this is only the case for an unrestricted number of clusters. If we restrict the number of clusters for hierarchical clustering to the number of categories, the differences become much less apparent (see tables 4 and 6). The only exception to this is marked by the tree edit distance: The best F-measure of 0.863 is achieved by using 58 clusters. If we restrict the number of clusters to 4, tree edit still reaches an F-measure of 0.710 which is significantly higher than the best k-means result of 0.599.

As one would intuitively expect the results achieved by the tree edit distance are much better than the variants of tree linearization. The edit distance operates on trees whereas the other algorithms are bound to less informative sequences. Interestingly, the differences become much less apparent for the corpus variant II which consists of the simplified DOM-trees (see section 3.2). We can assume that the advantage of the tree edit distance over the linearization-based approaches diminishes, the smaller the trees to be compared are.

The performance of the different variants of tree linearization vary only significantly in the case of unrestricted hierarchical clustering (see tables 3 and 5). In the case of k-means as well as in the case of restricting hierarchical clustering to exactly 4 clusters, the performances are about equal.

In order to provide a baseline for better rating the cluster results, we perform 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).

The baseline for the different variants of lin-

Similarity Algorithm	Clustering Algorithm	# Clusters	F-Measure	Purity	Inverse Purity	PW Distance	Method-Specifical
tree edit distance	hierarchical	58	0.863	0.996	0.786	none	weighted linkage
post-order linearization	hierarchical	13	0.775	0.809	0.775	spearman	single linkage
pre-order linearization	hierarchical	19	0.741	0.817	0.706	spearman	single linkage
tree level linearization	hierarchical	36	0.702	0.882	0.603	spearman	single linkage
bfs linearization	hierarchical	13	0.696	0.698	0.786	spearman	single linkage
tree edit distance	k-means	4	0.599	0.618	0.641	-	cosine distance
pre-order linearization	k-means	4	0.595	0.615	0.649	-	cosine distance
post-order linearization	k-means	4	0.593	0.615	0.656	-	cosine distance
tree level linearization	k-means	4	0.593	0.603	0.649	-	cosine distance
random lin. (medians only)	-	-	0.591	0.563	0.795	-	-
bfs linearization	k-means	4	0.580	0.595	0.656	-	cosine distance
-	random	4	0.311	0.362	0.312	-	-

Table 3: Evaluation results using all tags.

Similarity Algorithm	Clustering Algorithm	# Clusters	F-Measure	Purity	Inverse Purity	PW Distance	Method-Specifical
tree edit distance	hierarchical	4	0.710	0.698	0.851	spearman	single linkage
bfs linearization	hierarchical	4	0.599	0.565	0.851	none	weighted linkage
tree level linearization	hierarchical	4	0.597	0.615	0.676	spearman	complete linkage
post-order linearization	hierarchical	4	0.595	0.615	0.683	spearman	average linkage
pre-order linearization	hierarchical	4	0.578	0.599	0.660	cosine	average linkage

Table 4: Evaluation results using all tags and hierarchical clustering with a fixed number of clusters.

earization is given by random linearizations: We perform 16 random linearizations, run the different variants of distance measurement as well as clustering and compute the median of the best F-measure values achieved. These are 0.591 for corpus variant I and 0.581 for the simplified variant II. These results are in fact surprising because they are only little worse than the other linearization techniques. This result may indicate that – in the present scenario – the linearization based approaches to tree distance measurement are not suitable because of the loss of structure information. More specifically, this raises the following antithesis: Either, the sequence-oriented models of measuring structural similarities taken into account are insensitive to the structuring of web documents. Or: this structuring only counts what regards the degrees of nodes and their labels irrespective of their order. As tree-oriented methods perform better, we view this to be an argument against linearization oriented methods, at least what regards the present evaluation scenario *to which only DOM trees are input* but not more general graph structures.

The experiment has shown that analyzing the document structure provides a remarkable amount of information to categorization. It also shows that the sensitivity of the approaches used in different contexts needs to be further explored which we will address in our future research.

4 Conclusion

We presented a cluster-based approach to structure learning in the area of web documents. This was done in order to approach the goal of a combined algorithm of webgenre exploration *and* categorization. As argued in section (1), such an algorithm is needed in web corpus linguistics for webgenre tagging as a prerequisite of measuring genre-sensitive collocations. In order to evaluate the present approach, we utilized a corpus of wiki-based articles. The evaluation showed that there is an information gain when measuring the similarities of web documents irrespective of their lexical content. This is in the line of the genre model of systemic functional linguistics (Ventola, 1987) which prospects an impact of genre membership on text structure. As the corpus used for evaluation is limited to tree-like structures, this approach is in need for further development. Future work will address this task. This regards especially the classification of graph-like representations of web documents which take their link structure into account.

References

Einat Amitay, David Carmel, Adam Darlow, Ronny Lempel, and Aya Soffer. 2003. The connectivity sonar: detecting site functionality by structural patterns. In *Proc. of the 14th ACM conference on Hypertext and Hypermedia*, pages 38–47.

Marco Baroni and Silvia Bernardini, editors. 2006. *WaCky! Working papers on the Web as corpus.* Gedit, Bologna, Italy.

Philip Bille. 2003. Tree edit distance, alignment distance and inclusion. Technical report TR-2003-23.

Soumen Chakrabarti, Byron Dom, and Piotr Indyk.

Similarity Algorithm	Clustering Algorithm	# Clusters	F-Measure	Purity	Inverse Purity	PW Distance	Method-Specifical
tree edit distance	hierarchical	51	0.756	0.905	0.691	none	weighted linkage
pre-order linearization	hierarchical	20	0.742	0.809	0.771	spearman	single linkage
post-order linearization	hierarchical	23	0.732	0.813	0.756	spearman	single linkage
tree level linearization	hierarchical	2	0.607	0.553	0.878	spearman	weighted linkage
bfs linearization	hierarchical	4	0.589	0.603	0.641	cosine	weighted linkage
tree edit distance	k-means	4	0.713	0.718	0.718	-	cosine distance
pre-order linearization	k-means	4	0.587	0.603	0.634	-	cosine distance
tree level linearization	k-means	4	0.584	0.603	0.641	-	cosine distance
bfs linearization	k-means	4	0.583	0.599	0.637	-	cosine distance
post-order linearization	k-means	4	0.582	0.592	0.630	-	cosine distance
random lin. (medians only)	-	-	0.581	0.584	0.674	-	-
-	random	4	0.311	0.362	0.312	-	-

Table 5: Evaluation results using structure relevant tags only.

Similarity Algorithm	Clustering Algorithm	# Clusters	F-Measure	Purity	Inverse Purity	PW Distance	Method-Specifical
tree edit distance	hierarchical	4	0.643	0.645	0.793	spearman	average linkage
post-order linearization	hierarchical	4	0.629	0.634	0.664	spearman	average linkage
tree level linearization	hierarchical	4	0.607	0.595	0.679	spearman	weighted linkage
bfs linearization	hierarchical	4	0.589	0.603	0.641	cosine	weighted linkage
pre-order linearization	hierarchical	4	0.586	0.603	0.660	cosine	complete linkage

Table 6: Evaluation results using all tags and hierarchical clustering with a fixed number of clusters.

1998. Enhanced hypertext categorization using hyperlinks. In *Proceedings of ACM SIGMOD International Conference on Management of Data*, pages 307–318. ACM.

Weimin Chen. 2001. New algorithm for ordered tree-to-tree correction problem. *Journal of Algorithms*, 40(2):135–158.

Matthias Dehmer. 2005. *Strukturelle Analyse Web-basierter Dokumente*. Ph.D. thesis, Technische Universität Darmstadt, Fachbereich Informatik.

Johannes Fürnkranz. 1999. Exploiting structural information for text classification on the WWW. In *Proceedings of the Third International Symposium on Advances in Intelligent Data Analysis*, pages 487–498, Berlin/New York. Springer.

M. Höchsmann, T. Töller, R. Giegerich, and S. Kurtz. 2003. Local similarity in rna secondary structures. In *Proc. Computational Systems Bioinformatics*, pages 159–168.

Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin. 2003. A practical guide to SVM classification. Technical report, Department of Computer Science and Information Technology, National Taiwan University.

Thorsten Joachims. 2002. *Learning to classify text using support vector machines*. Kluwer, Boston.

Adam Kilgarriff and Gregory Grefenstette. 2003. Introduction to the special issue on the web as corpus. *Computational Linguistics*, 29(3):333–347.

P. Klein. 1998. Computing the edit-distance between unrooted ordered trees. In G. Bilardi, G. F. Italiano, A. Pietracaprina, and G. Pucci, editors, *Proceedings of the 6th Annual European Symposium*, pages 91–102, Berlin. Springer.

Jon M. Kleinberg. 1999. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632.

Alexander Mehler and Rüdiger Gleim. 2005. The net for the graphs — towards webgenre representation for corpus linguistic studies. In Marco Baroni and Silvia Bernardini, editors, *WaCky! Working papers on the Web as corpus*. Gedit, Bologna, Italy.

Alexander Mehler, Rüdiger Gleim, and Matthias Dehmer. 2005. Towards structure-sensitive hypertext categorization. In *Proceedings of the 29th Annual Conference of the German Classification Society*, Berlin. Springer.

Yoshiaki Mizuuchi and Keishi Tajima. 1999. Finding context paths for web pages. In *Proceedings of the 10th ACM Conference on Hypertext and Hypermedia*, pages 13–22.

Michael Stubbs. 2001. On inference theories and code theories: Corpus evidence for semantic schemas. *Text*, 21(3):437–465.

K. C. Tai. 1979. The tree-to-tree correction problem. *Journal of the ACM*, 26(3):422–433.

Eija Ventola. 1987. *The Structure of Social Interaction: a Systemic Approach to the Semiotics of Service Encounters*. Pinter, London.

Yiming Yang, Sean Slattery, and Rayid Ghani. 2002. A study of approaches to hypertext categorization. *Journal of Intelligent Information Systems*, 18(2-3):219–241.

Yiming Yang. 1999. An evaluation of statistical approaches to text categorization. *Journal of Information Retrieval*, 1(1/2):67–88.

K. Zhang and D. Shasha. 1989. Simple fast algorithms for the editing distance between trees and related problems. *SIAM Journal of Computing*, 18:1245–1262.