SITEGUIDE: AN EXAMPLE-BASED APPROACH TO WEB SITE DEVELOPMENT ASSISTANCE

Vera Hollink, Viktor de Boer and Maarten van Someren
University of Amsterdam, Science Park 107, Amsterdam, The Netherlands

Keywords: Information architecture, Web site modeling, Web site design support.

Abstract: We present 'SiteGuide', a tool that helps web designers to decide which information will be included in a new web site and how the information will be organized. SiteGuide takes as input URLs of web sites from the same domain as the site the user wants to create. It automatically searches the pages of these example sites for common topics and common structural features. On the basis of these commonalities it creates a model of the example sites. The model can serve as a starting point for the new web site. Also, it can be used to check whether important elements are missing in a concept version of the new site. Evaluation shows that SiteGuide is able to detect a large part of the common topics in example sites and to present these topics in an understandable form to its users.

1 INTRODUCTION

Even the smallest companies, institutes and associations are expected to have their own web sites. However, designing a web site is a difficult and time-consuming task. Software tools that provide assistance for the web design process can help both amateur and professional web designers.

Newman and Landay (2000) studied the current practices in web design and identified four main phases in the design process of a web site: discovery, design exploration, design refinement and production. A number of existing tools, such as Adobe Dreamweaver[1] and Microsoft Frontpage[2] provide help for the latter two phases, where an initial design is refined and implemented. These tools however, do not support collecting and structuring the content into an initial conceptual model (Falkovych and Nack, 2006). In this paper, we present 'SiteGuide', a system that helps web designers to create a setup for a new site. Its output is an initial information architecture for the target web site that shows the user what information should be included in the website and how the information should be structured. Figure 1 shows a screenshot of the SiteGuide system.

An important step in the discovery phase of web site design is reviewing web sites from the same domain as the target site (Newman and Landay, 2000). For instance, a person who wants to build a site for a small soccer club will often look at web sites of some other small soccer clubs. The information architectures of the examined sites are used as source of inspiration for the new site.

Reviewing example sites can provide useful information, but comparing sites manually is very time-consuming and error-prone, especially when the sites consist of many pages. The SiteGuide system creates an initial information architecture for a new site by efficiently and systematically comparing a set of example sites identified by the user. SiteGuide automatically searches the sites for topics and structures that the sites have in common. For example, in the soccer club domain, it may find that most example sites contain information about youth teams or that pages about membership always link to pages about subscription fees. The common topics are brought together in a model of the example sites. The model is presented to the user and serves as an information architecture for the new web site.

SiteGuide can also be used in the design refinement phase of the web design process as a critic of a first draft of a site. The draft is compared with the model, so that missing topics or unusual information structures are revealed.

[1] http://www.adobe.com/products/dreamweaver
[2] http://office.microsoft.com/frontpage

Hollink V., de Boer V. and van Someren M.
SITEGUIDE: AN EXAMPLE-BASED APPROACH TO WEB SITE DEVELOPMENT ASSISTANCE.
DOI: 10.5220/0001825401430150
In *Proceedings of the Fifth International Conference on Web Information Systems and Technologies (WEBIST 2009)*, page
ISBN: 978-989-8111-81-4

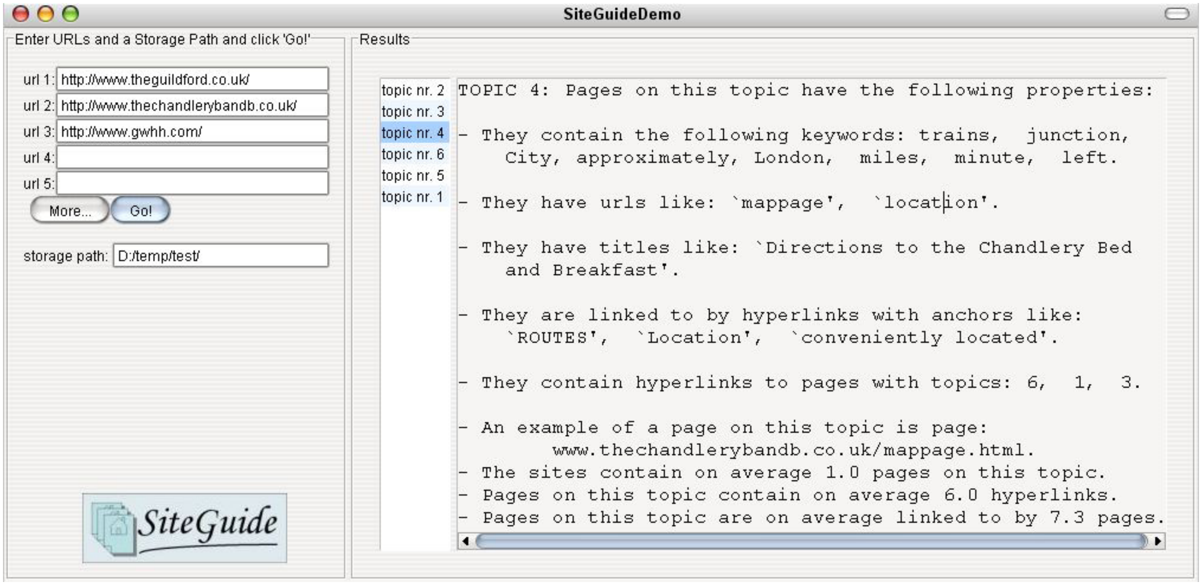

Figure 1: A screenshot of the SiteGuide system showing a topic for a web site of a hotel.

2 PROBLEM DEFINITION

The SiteGuide system has two main usage scenarios, shown in Figure 2. In both scenarios the user starts the interaction by inputting the URLs of the home pages of a small set of example web sites. SiteGuide then scrapes and analyzes the sites and captures their commonalities in a web site model. The model forms the suggested information architecture for the new site. In the *modeling scenario* the information architecture is the end point of the interaction and is outputted to the user. In the *critiquing scenario* the user has already created a first draft version of his new site. SiteGuide compares the draft with the model of the example sites and outputs the differences.

Figure 3 shows the structure of an example site model. A model consists of a set of *topics* that appear in the example sites. To communicate the model to a user, SiteGuide describes each topic with a set of *characterizing features*. These features explain to the user what the topic is about. They consist of key

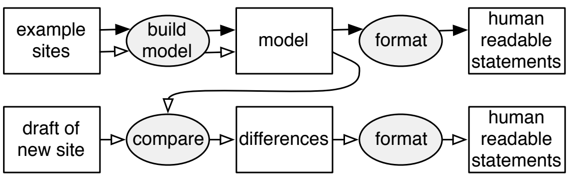

Figure 2: The two usage scenarios of the SiteGuide system. —► denotes the modeling scenario. —▷ denotes the critiquing scenario.

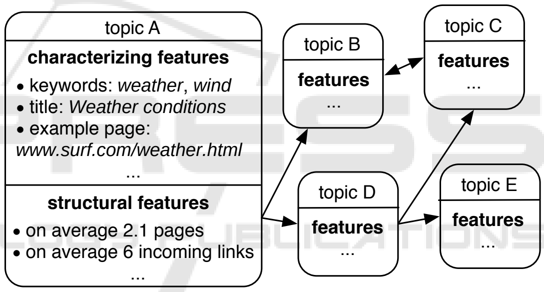

Figure 3: Example of an example site model. A model consists of topics that have characterizing and structural features (only shown for topic A). Frequently occurring hyperlinks between topics are denoted by arrows.

phrases which are extracted from the contents of the pages that handle on the topic as well as titles of these pages, anchor texts of links pointing to the pages and terms from the page URLs. Additionally, SiteGuide shows a link to a page that exemplifies the topic.

To inform the user on how a topic should be embedded in the site, SiteGuide shows *structural features* that describe how the topic is represented in the pages of the example sites. It shows the average number of pages about the topic, the average number of incoming and outgoing links for those pages and links between topics (e.g., pages on topic A frequently link to pages on topic B).

In the modeling scenario the topics of the model are presented to the user as a set of natural language statements. The screenshot in Figure 1 shows the current visualization of the output of a topic. The information architecture is also exported to an XML file, so that it can be imported into other web design tools that provide alternative visualizations, such as prototyping or wireframing tools.

The output of the critiquing scenario is a set of similar statements that describe the differences between the example sites and the draft. In this scenario SiteGuide indicates which topics are found on most example sites but not on the draft site and which topics are found on the draft site but not in any of the example sites. In addition, it outputs topics that are both on the draft and the example sites, but that have different structural features.

3 METHOD

The main task of SiteGuide is to find topics that occur on most example sites. For this, SiteGuide identifies pages of different example sites that handle on the same topic and maps these pages onto each other. A mapping can be seen as a set of page clusters. Each cluster contains all pages from example sites that handle on one topic.

We design our clustering format in such a way that it is able to capture differences between web sites in the information they present and in the way the information is organized. All pages of the example sites must occur in at least one page cluster, but pages may occur in multiple clusters. In the simplest case a cluster contains one page from each example site. For example, a cluster can contain for each site the page about surfing lessons. However, if on one of the sites the information about surfing lessons is split over several pages, these pages are placed in the same cluster. It can also happen that a cluster does not contain pages from all sites, because some of the sites do not contain information on the cluster's topic. Pages occur in more than one cluster, when they contain content about more than one topic.

We developed a heuristic method to find a good mapping between a set of example sites. A space of possible mappings is defined and searched for a good mapping. Below we first explain how SiteGuide measures the quality of a mapping. Then we explain how it searches for the mapping that maximizes the quality measure. Finally, the generation of the example site model from the mapping and the comparison between the model and a draft of the new site are discussed.

3.1 Quality Measure

As in most clustering tasks, the quality of an example site mapping is better when the pages in the clusters are more similar to each other. However, in most domains pages of one site are more similar to pages on the same site that handle on other topics than to pages of other sites on the same topic. As a result, a standard clustering method would mainly form groups of pages from one example site instead of identifying topics that span multiple sites. We solve this problem by focussing on similarities between pages from different sites. We define the quality of a page cluster as the average similarity of the pages in the cluster to all other pages in the cluster *from other web sites*.

Most web pages contain some text, so that text similarity measures can be used to compute the similarity between two pages. However, web pages are not stand-alone texts, but part of a network that is connected by links. In SiteGuide we make use of the information contained in the link structure by computing the similarity between the pages' positions in their link structures. As extra features we use the similarity between page titles, page URL's and the anchors of the links that point to pages. Below, each of these five types of similarity are discussed in more detail.

The quality of a page cluster is a combination of the five similarity measures. The quality of cluster C in mapping M is:

$$quality(C,M) = \sum_{sim_i \in Sims} (w_i \cdot sim_i(C,M)) - \alpha \cdot S_C$$

Here *Sims* are the five similarity measures, which are weighted with weighting parameters w_i. S_C is the number of example sites that have pages in cluster C. α is a parameter.

The term $-\alpha \cdot S_C$ subtracts a fixed amount (α) for each of the S_C sites in the cluster. Consequently, adding pages of a site to a cluster only improves the cluster's score if the pages bear a similarity of at least α to the pages of the other sites in the cluster. In this way the size of the clusters is automatically geared to the number of sites that address the same topic, so that we do not need to specify the number of clusters beforehand.

Text similarity between two pages is expressed as the cosine similarity between the terms on the pages (Salton and McGill, 1983). This measure enables SiteGuide to identify parts of the texts that pages have in common and ignore site-specific parts. Stop word removal, stemming and $tf \cdot idf$ weighting are applied to increase accuracy.

Anchor text similarity between two pages is defined as the cosine similarity between the anchor texts of the links that point to the pages. For the computation of page title similarity and URL similarity we use the Levenshtein distance (Levenshtein, 1966) instead of the cosine similarity. Levenshtein distance is more suitable for comparing short phrases as it takes the order of terms into account and works at character level instead of term level.

We developed a new measure to compute the similarity between the positions of two pages in their link structures. We look at the direct neighborhood of each page: it's incoming and outgoing links. The link structure similarity of a cluster is the proportion of the incoming and outgoing links of the pages that are mapped correctly. Two links in different link structures are mapped correctly onto each other if both their source pages and their destination pages are mapped onto each other.

3.2 Finding a Good Mapping

A naive approach for finding a mapping with a high quality score would be to list all possible mappings, compute for each mapping the quality score, and choose the one with the highest score. Unfortunately, this approach is not feasible, as the number of possible mappings is extremely large (Hollink et al., 2008). To make the problem computationally feasible, we developed a search space of possible mappings that allows us to heuristically search for a good mapping.

We start our search with an initial mapping that is likely to be close to the optimal solution. In this mapping each page occurs in exactly one cluster and each cluster contains no more than one page from each example site. The initial mapping is built incrementally. First, we create a mapping between the first and the second example site. For each two pages of these sites we compute the similarity score defined above. The so called Hungarian Algorithm (Munkres, 1957) is applied to the two sites to find the one-to-one mapping with the highest similarity. Then, the pages of the third site are added to the mapping. We compute the similarity between all pages of the third site and the already formed pairs of pages of the first two sites and again apply the Hungarian Algorithm. This process is continued until all example sites are included in the initial mapping.

We define five mapping modification operations that can be used to traverse the search space. Together, these operations suffice to transform any mapping into any other mapping. This means that the whole space of possible mappings is reachable from any starting point. The operations are:

- Split a cluster: the pages from each site in the cluster are placed in a separate cluster.
- Merge two clusters: place all pages from the two clusters in one cluster.
- Move a page from one cluster to another cluster.
- Move a page from a cluster to a new, empty cluster.
- Copy a page from one cluster to another cluster.

With these operations SiteGuide refines the initially created mapping using a form of hill climbing. In each step it applies the operations to the current mapping and computes the effect on the similarity score. When an operation improves the score it is retained; otherwise it is undone. It keeps trying modification operations until it can not find any more operations that improve the score with a sufficient amount.

The five operations can be applied to all clusters. To increase efficiency, SiteGuide tries to improve clusters with low quality scores first.

3.3 From Mapping to Model

The next step is to transform the mapping into a model of the example sites. The mapping consists of page clusters, while the model should consist of descriptions of topics that occur on most of the example sites.

Each cluster becomes a topic in the model. Topics are characterized by the five characterizing features mentioned in Section 2. SiteGuide lists all terms from the contents of the pages and all URLs, titles and anchor texts. For each type of term we designed a measure that indicates how descriptive the term or phrase is for the topic. For instance, content terms receive a high score if they occur in all example sites frequently in pages on the topic and infrequently on other pages. These scores are multiplied by the corresponding similarity scores, e.g., the score of a content term is multiplied by the topic's content similarity. The result of this is that features that are more important for a topic are weighted more heavily. The terms and phrases with scores above some threshold (typically 3 to 10 per feature type) become characterizing features for the topic. The most central page in the cluster (the page with the highest text similarity to the other pages in the cluster) becomes the example page for the topic.

To find the structural features of the topics, SiteGuide analyzes the pages and links of the corresponding page clusters. It determines for each site over how many pages the information on a topic is spread and counts the number of incoming and outgoing links. Furthermore, it counts how often the topic links to each other topic. The question is which of these numbers indicate a stable pattern over the various example sites. Intuitively, we recognize a pattern in, for instance, the number of outgoing links of a topic, when on all sites the pages on the topic have roughly the same number of outgoing links. We have formalized this intuition: when the numbers for the various sites have low variance, SiteGuide marks the feature as a common structural feature.

In the current version of SiteGuide the output of the modeling scenario consists of a series of human readable statements (see Figure 1). For each topic SiteGuide outputs the characterizing features, the example page and the common structural features. In the next version, the model will be shown graphically, more or less like Figure 3.

For the critiquing scenario we developed a variant of the web site comparison method which enables SiteGuide to compare the example site model to a draft of the new site. This variant uses the hill climbing approach to map the model to the draft, but it does not allow any operations that alter the example site model. In this way we ensure that the draft is mapped onto the model, while the model stays intact. Once the draft is mapped, SiteGuide searches for differences between the draft and the example site model. It determines which topics in the model do not have corresponding pages in the draft and reports that these topics are missing on the new site. Conversely, it determines which topics of the draft do not have counterparts in the example sites. Finally, it compares the structural features of the topics in the new site to the common structural features in the example site model and reports the differences.

4 EVALUATION

To determine whether SiteGuide can provide useful assistance to users who are building a web site, we need to answer two questions. 1) Do the discovered topics represent the subjects that are really addressed at the example sites? 2) Are the topic descriptions understandable for humans? To answer the first question we compared mappings created by SiteGuide to manually created example site mappings. For the second question we asked humans to interpret SiteGuide's output.

We used web sites from three domains: windsurf clubs, primary schools and small hotels. For each domain 5 sites were selected as example sites. We purposely chose very different domains: the windsurf clubs are non-profit organizations, the school domain is an educational domain and the hotel domain is commercial. Table 1 shows the main properties of the three domains.

The SiteGuide system generated example site models for the three domains. We compared these models to gold standard models that we had constructed by hand. The gold standard models consisted of a mapping between the example sites and for each page cluster a textual description of the topic that was represented by the cluster. The textual descriptions were 1 or 2 sentences in length and contained around 20 words. For example, in the school domain one topic was described as '*These pages contain a list of staff members of the schools. The lists consist of the names and roles of the staff members.*'. Features of the gold standards are given in Table 1.

Table 1: Properties of the evaluation domains and the gold standards (g.s.): the total, minimum and maximum number of pages in the example sites, the number of topics in the g.s., the number of topics that were found in at least 50% of the sites (frequent topics) and the percentage of the pages that were mapped onto at least one other page.

domain	total pages	min-max pages	topics in g.s	frequent topics in g.s.	% pages mapped in g.s.
hotel	59	9-16	21	7	81%
surfing	120	8-61	90	12	76%
school	154	20-37	42	17	80%

To validate the objectivity of the gold standards, for one domain (hotels) we asked another person to create a second gold standard. We compared the topics in the two gold standards and found that 82% of the topics were found in both gold standards. From this we concluded that the gold standards were quite objective and are an adequate means to evaluate the output of SiteGuide.

4.1 Evaluation of the Mappings

First, we evaluated the page clusters that were generated by SiteGuide to see to what extend these clusters coincided with the manually created clusters in the gold standards, in other words, to what extend they represented topics that were really addressed at the example sites. We counted how many of the clusters in the generated mapping also occurred in the gold standards. A generated cluster was considered to be the same as a cluster from the gold standard if at least 50% of the pages in the clusters were the same. We considered only topics that occurred in at least half of the example sites (frequent topics), as these are the topics that were present on most of the example sites.

The quality of mappings is expressed by precision, recall and f-measure over the page clusters. When C_{gold} are the clusters in a gold standard and C_{test} are the clusters in a generated mapping, the measures are defined as:

$$\text{precision} = |C_{test} \cap C_{gold}| / |C_{test}|$$
$$\text{recall} = |C_{test} \cap C_{gold}| / |C_{gold}|$$
$$\text{f–measure} = \frac{(2 \cdot \text{precision} \cdot \text{recall})}{(\text{precision} + \text{recall})}$$

In a pilot study (Hollink et al., 2008) we tested the influence of the various parameters. We generated example site mappings with various weights for

Table 2: Quality of the example site mappings and quality of the comparison between drafts and example sites.

domain	precision	recall	f-measure	removed topics detected	added topics detected
hotel	1.00	0.43	0.60	0.54	0.52
surf	0.33	0.42	0.37	0.25	0.26
school	0.47	0.41	0.44	0.33	0.88

the similarity measures. On all three domains giving high weights to text similarity resulted in mappings with high scores. In the hotel domain URL similarity also appeared to be effective. Increasing the minimum similarity parameter (α) meant that we required mapped pages to be more similar, so that precision was increased, but recall decreased. Thus, with this parameter we can effectively balance the quality of the topics that we find against the number of topics that are found. When the SiteGuide system is used by a real user, it obviously cannot use a gold standard to find the optimal parameter settings. Fortunately, we can estimate roughly how we should choose the parameter values by looking at the resulting mappings as explained in (Hollink et al., 2008).

The scores of the example site models generated with optimal parameter values are shown in Table 2. The table shows the scores for the situation in which all frequent topics that SiteGuide has found are shown to the user. When many topics have been found we can choose to show only topics with a similarity score above some threshold. In general, this improves precision, but reduces recall.

Next, we evaluated SiteGuide in the critiquing scenario. We performed a series of experiments in which the 5 sites one by one played the role of the draft site and the remaining 4 sites were example sites. In each run we removed all pages about one of the gold standard topics from the draft site and used SiteGuide to compare the corrupted draft to the examples. We counted how many of the removed topics were identified by SiteGuide as topics that were missing in the draft. Similarly, we added pages to the draft that were not relevant in the domain. Again, SiteGuide compared the corrupted draft to the examples. We counted how many of the added topics were marked as topics that occurred only on the draft site and not on any of the example sites. The results are given in Table 2.

Table 2 shows that SiteGuide is able to discover many of the topics that the sites have in common, but also misses a number of topics. Inspection of the created mappings demonstrates that many of the discovered topics can indeed lead to useful recommendations to the user. We give a few examples. In the school domain SiteGuide created a page cluster that contained for each site the pages with term dates. It also found correctly that 4 out of 5 sites provided a list of staff members. In the surfing domain, a cluster was created that represented pages where members could leave messages (forums). The hotel site mapping contained a cluster with pages about the facilities in the hotel rooms. The clusters can also be relevant for the critiquing scenario: for example, when the owner of the fifth school site would use SiteGuide, he would learn that his site is the only site without a staff list.

Some topics that the sites had in common were not found, because the terms did not match. For instance, two school sites provided information about school uniforms, but on the one site these were called ‘uniform’ and on the other ‘school dress’. This example illustrates the limitations of the term-based approach. In the future, we will extend SiteGuide with WordNet (Fellbaum, 1998), which will enable it to recognize semantically related terms.

4.2 Evaluation of the Topic Descriptions

Until now we counted how many of the generated topics had at least 50% of their pages in common with a gold standard topic. However, there is no guarantee that the statements that SiteGuide outputs about these topics are really understood correctly by the users of the SiteGuide system. Generating understandable descriptions is not trivial, as most topics consist of only a few pages. On the other hand, it may happen that a description of a topic with less than 50% overlap with a gold standard topic is still recognizable for humans. Therefore, below we evaluate how SiteGuide's end output is interpreted by human users and whether the interpretations correspond to the gold standards.

We used SiteGuide to create output about the example site models generated with the same optimal parameter values as in the previous section. Since we only wanted to evaluate how well the topics could be interpreted by a user, we did not output the structural features. We restricted the output for a topic to up to 10 content keywords and up to 3 phrases for page titles, URLs and anchor texts. We also displayed for each topic a link to the example page.

Output was generated for each of the 34 frequent topics identified in the three domains. We asked 5 evaluators to interpreted the 34 topics and to write a short description of what they thought each topic was about. We required the descriptions to be of the same length as the gold standard descriptions (10-30 words). None of the evaluators were domain experts or expert web site builders. It took the evaluators on average one minute to describe a topic, including typing the description. By comparison, finding the topics in the example sites by hand (for the creation of the gold standard) took about 15-30 minutes per topic.

An expert coder determined whether the interpretations of the evaluators described the same topics as the gold standard descriptions. Since both the evaluators' topic descriptions and the gold standard topic descriptions were natural language sentences, it was often difficult to determine whether two descriptions described the exact same topic. We therefore had the coder classify each description in one of three classes: a description could either have a partial match or an exact match with one of the gold standard topics or have no matching gold standard topic at all. An exact match means that the description describes the exact same topic as the gold standard topic. A partial match occurs when a description describes for instance a broader or narrower topic. To determine precision the coder matched all topic descriptions of the evaluators to the gold standard descriptions. To determine recall the coder matched all gold standard descriptions to the evaluators' descriptions in the same manner. To determine the objectivity of the coding task, we had a second expert coder perform the same task. The two coders agreed on 69% of the matches, considering the possible variety in topic descriptions, we consider this an acceptable level of inter-coder agreement.

Averaged over the domains and evaluators, 94% of all evaluators' topic descriptions were matched to one of the gold standard topics when both partial and exact matches were considered. In other words, 94% of the topics that were found by SiteGuide corresponded (at least partially) with a topic from the gold standard and were also interpreted as such. When only exact matches were counted this figure was still 73%. This indicates that for most topics SiteGuide is capable of generating an understandable description.

We calculated precision and recall based on the relaxed precision and recall used in ontology matching literature (Ehrig and Euzenat, 2005), resulting in the same formula for precision and recall as in Section 4.1, where $|\mathcal{C}_{test} \cap \mathcal{C}_{gold}|$ is the number of exact matches plus 0.5 times the number of partial matches. In Table 3 we display the relaxed precision, recall and f-measure values for the three domains. The average precision over all domains is 0.83, showing that most of the topics that SiteGuide found could indeed be interpreted and that these interpretations corresponded to correct topics according to our gold standard. The average recall is 0.57, which means that more than half of the topics from the gold standard were correctly identified and outputted by SiteGuide. Both precision and recall do not vary much across domains, indicating that SiteGuide is capable of identifying and displaying topics in a wide range of domains. The results in Table 3 are considerably better than those in Table 2. This shows that for a number of topics where the page overlap with the gold standard was less than 50%, the displayed topic could still be interpreted correctly by humans.

Table 3: Results of the manual evaluation for the three domains.

domain	precision	recall	f-measure
hotel	0.90	0.54	0.68
school	0.78	0.54	0.64
surf	0.81	0.63	0.71

5 RELATED WORK

Existing tools for assisting web site development help users with the technical construction of a site. Tools such as Dreamweaver[1] or Frontpage[2] allow users to create web sites without typing HTML. Other tools evaluate the design and layout of a site on usability and accessibility, checking for example for dead links and buttons and missing captions of figures (see for an overview (Web Accessibility Initiative, 2008; Brajnik, 2004)). However, none of these tools help users to choose appropriate contents or structures for their sites or critique the content that is currently used.

Our approach is in spirit related to the idea of ontologies. The goal of an ontology is to capture the conceptual content of a domain. It consists of structured, formalized information on the domain. The web site models presented in this paper can be viewed as informal ontologies as they comprise structures of topics that occur in sites from some domain. However, our models are not constructed by human experts, but automatically extracted from example sites.

Another related set of tools are tools that improve link structures of web sites, such as PageGather (Perkowitz and Etzioni, 2000) and the menu optimization system developed by Hollink et al. (Hollink et al., 2007). These tools do not provide support on the contents of a site. Moreover, they need usage data, which means that they can only give advice about sites that have been online for some time.

The algorithm that underlies the SiteGuide system is related to methods for high-dimensional clustering, which are, for instance, used for document clustering. However, there are several important differences. The task of SiteGuide is to find topics in a set of web sites instead of an unstructured set of documents. It is more important to find topics that appear in many sites than to group pages within sites. This is reflected in the similarity measure. Another difference is that in SiteGuide the relations (links) between pages are taken into account. The extent to which relations between pages within one site match the relations in other sites contributes to the similarity between pages. Also, the final model of the sites includes relations between topics.

6 CONCLUSIONS

The SiteGuide system provides assistance to web designers who want to build a web site but do not know exactly which content must be included in the site. It automatically compares a number of example web sites and constructs a model that describes the features that the sites have in common. The model can be used as an information architecture for the new site. In addition, SiteGuide can show differences between example sites and a first version of a new site.

SiteGuide was applied to example web sites from three domains. In these experiments, SiteGuide proved able to find many topics that the sites had in common. Moreover, the topics were presented in such a way that humans could easily and quickly understand what the topics were about.

Although the results of the evaluation are promising, user experiments are needed to test the value of SiteGuide in practice. In an experiment we will ask users to design a web site. Half of the users will perform this task with the help of SiteGuide and the other half without. Comparison of the time needed for the design and the quality of the resulting drafts will show how useful the system is in practice.

In the near future SiteGuide will be extended with a number of new features. Semantics will be added to the similarity measure in the form of WordNet relations. We will connect the output to prototyping tools, so that users can directly start editing the proposed site design. Finally, SiteGuide could output additional features such as style features (e.g., colors and use of images) or the amount of tables, lists, forms, etc. More research is needed to determine how these features can be compared automatically.

REFERENCES

Brajnik, G. (2004). Using automatic tools in accessibility and usability assurance processes. In *Proceedings of the Eighth ERCIM Workshop on User Interfaces for All, Vienna, Austria*, pages 219–234.

Ehrig, M. and Euzenat, J. (2005). Relaxed precision and recall for ontology matching. In *Proceedings of the K-Cap 2005 Workshop on Integrating Ontologies, Banff, Canada*, pages 25–32.

Falkovych, K. and Nack, F. (2006). Context aware guidance for multimedia authoring: Harmonizing domain and discourse knowledge. *Multimedia Systems*, 11(3):226–235.

Fellbaum, editor, C. (1998). *WordNet: An electronic lexical database*. MIT Press, Cambridge, MA.

Hollink, V., Van Someren, M., and De Boer, V. (2008). Capturing the needs of amateur web designers by means of examples. In *Proceedings of the 16th Workshop on Adaptivity and User Modeling in Interactive Systems, Würzburg, Germany*, pages 26–31.

Hollink, V., Van Someren, M., and Wielinga, B. (2007). Navigation behavior models for link structure optimization. *User Modeling and User-Adapted Interaction*, 17(4):339–377.

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 10(8):707–710.

Munkres, J. (1957). Algorithms for the assignment and transportation problems. *Journal of the Society of Industrial and Applied Mathematics*, 5(1):32–38.

Newman, M. W. and Landay, J. A. (2000). Sitemaps, storyboards, and specifications: A sketch of web site design practice. In *Proceedings of the Third Conference on Designing Interactive Systems, New York, NY*, pages 263–274.

Perkowitz, M. and Etzioni, O. (2000). Towards adaptive web sites: Conceptual framework and case study. *Artificial Intelligence*, 118(1-2):245–275.

Salton, G. and McGill, M. J. (1983). *Introduction to Modern Information Retrieval*. McGraw-Hill, New York, NY.

Web Accessibility Initiative (2008). Web accessibility evaluation tools: Overview. *http://www.w3.org/WAI/ER/tools/*. Last accessed: July 4, 2008.