Contextual Rearrangement of Web Content

Mario Kubek, Herwig Unger, Pongpol Hussein, Wairung Tiranalinvit, and Paphawee Chatpaiboonwat

Abstract—In the future, workflows will be an important virtual good to trade with. Nowadays, web applications realise fixed workflows such as booking a flight or train ticket and web documents often provide step-by-step instructions to users in order to reach a goal. The automatic construction and acquisition of workflows is currently not possible. In this paper, new solutions are described to rearrange web contents even across the borders of their web documents. This makes it possible for users to provide context-dependent user feedback in order to make other users aware of topically connected contents. These solutions can be regarded as a first step to construct workflows by evaluating this feedback of web users.

Keywords—workflow generation, web content analysis, source topic detection, link induction, web browser extension

I. INTRODUCTION

In the next years, context-depending workflows will play an important role as virtual goods. That means that sequences of activities can be suggested to and shared by users in need in order to reach a goal in a specific situation. Hereby, the question of ‘What is being dealt with?’ plays an important, yet secondary role; the question of ‘How to do it?’, in contrast, is of main interest. However, the automatic extraction or construction of workflows is a hard task, when connected information is not available for this purpose. Also, it needs to be regarded that workflows are carried out in and are bound to special contexts (e.g. locations, times, tasks at hand, specific input and output parameters).

Web documents such as web pages present a valuable source and the context for up-to-date information on topics of interest. Also, they can be regarded as a means to connect the real with the virtual world because web documents often are representations of real-world entities (e.g. hotels or restaurants) and can explain courses of action in real-world situations. In each case, this information refers to a specific context in which actions can occur (e.g. renting a car and driving to the hotel for the context “arrival at the destination airport”). However, a complete workflow for a specific context might not be available on one particular source.

Instead, information to construct a usable workflow can be widespread on different sources, especially when dealing with similar contexts. Users, however, can easily determine topically connected information that is relevant for specific contexts based on their experiences. Therefore, it is sensible to actively involve users to generate or compile workflows by interactively rearranging, spreading and combining connected information while they browse through web contents. This way, the users themselves become a part of a specific context as their knowledge is used actively to reshape the content.

However, the problem at this is that although web pages can be dynamically created, they usually present static content (apart from web pages that rely on Ajax technology such as web-based e-mail and chat systems that constantly retrieve updated information) that is published and managed by its administrator. In each case, the web user has no or only very limited means to extend or change the presented content once it is loaded by the used web browser. This fact actually hinders users to actively participate in shaping the web according to their needs and wishes. Even existing web annotation tools such as [1] and [2] do not provide enough flexibility to actually (re-)use web content in a different context such as in another web page. Although, this use of content might not be intended by the administrators, for the users, it presents a valuable option to combine web contents when it seems appropriate to them. This way, a real dynamisation of web contents can be reached that actively involves the user community and is of benefit for it. This dynamic content aggregation from different sources is thus a requirement for the generation of workflows and ultimately their distribution as virtual goods by the users.

Therefore, in this paper, technical solutions are introduced to rearrange web content even across the borders of its web documents. This makes it possible for users to provide context-depending user feedback in order to make other users aware of topically connected content. The transfer of content involves and includes the following elements:

- textual information, pictures and animations
- formatting instructions in HTML and code fragments (e.g. in JavaScript)
- links to other documents in the content
- a link to the content’s source document

The technical solutions are based on client-sided extensions of web browsers as thin clients that capture the user feedback and provide its graphical presentation and server-sided content and user management facilities.

These solutions can be regarded as a first step to explicitly construct user-generated workflows without even the need to
manually enter any text such as comments or annotations. Also, in future applications of this user feedback, it is possible to analyse and evaluate it automatically in order to derive context-dependent as well as topically and semantically ordered workflows.

The remaining paper is structured as follows: the next section explains the methodology used and the technical prerequisites to realise these solutions. Section three introduces the actual technical implementations and explains the supported use cases. In section four, further application scenarios such as the automatic analysis of user feedback to generate workflows and the possibility to recommend topically matching web content to users using automatic link induction based on content analysis are discussed in detail. Section five concludes the paper and provides a look at future options to further enhance the introduced solutions, e.g. to actively support the user in creating workflows based on existing web content and its semantic relations.

II. METHODOLOGY

A. Technical Prerequisites

The most important tool for the user to deal with web content in the described manner is of course the web browser. However, as no current web browser currently offers the functionality to rearrange web content by itself, it is needed to extend it accordingly. Web browsers such as Google Chrome or Firefox, however, offer application programming interfaces (APIs) to add extra functionality to those browsers through extensions. Therefore, it is sensible to realise the described function to rearrange web content using browser extensions that can be seamlessly integrated in the graphical user interface (GUI) of the particular browser.

The other important part of such a realisation is the server part which is responsible for the user and content management. The main task of the server hereby is to realise a mapping of rearranged web content from different sources to the correct web content on the destination web page and to make this rearrangement persistent for each user using an attached database. Also, the server has to deliver the changes made to the invoked web page by a web browser that are then interpreted by its installed extension and applied to the original web page. This way, the original web page will not be changed, but additional content will be made instantly available, when the user calls the page.

B. Contextual Management of Web Content

As outlined in the introduction, the aim of the efforts presented herein is to present technical means to

- select specific web content,
- rearrange this content on its web document or to attach it to web content on other web documents
- and to present these changes and modifications to users when they call web documents enriched this way.

In other words, users should be able to use content from a specific context (e.g. webpage A) in a new context (e.g. webpage B) where it seems appropriate. This way, several (fragments of) contexts can be merged by user feedback in order to make semantically connected contents visible to users. However, the content is only one contextual dimension in these considerations. Another one is the user who is actually providing valuable feedback by this means. In particular, besides the content (including formatting instructions using HTML tags and JavaScript code) transferred, the following kind of information can be derived from this feedback:

- the username
- date and time
- URL of the source web document
- web browser used

With this information, it is possible to present the enriched content of a web document using additional filters, e.g. to only load and show modifications of specific users. Filtering is a very important aspect to prevent information overload, when too many feedback information would be presented at once. In this case, the original content might be visually covered by feedback information of many users that might draw the attention of the users to a great degree, which is definitely undesirable. Therefore, it is necessary to provide a useful solution to filter and (re)order attached content. The most important aspect hereby, however, is to topically group the contents provided by many users for information condensation purposes. This task is challenging as the aggregated contents from other web pages can cover many different topics of different relevance to the content they are attached to.

Therefore, it is sensible to employ special clustering techniques in order to find not only topical similarities but also dependencies in the user feedback and to determine its relatedness to the enriched web content. In [3], a graph-based clustering algorithm has been introduced to determine the topical sources of text documents. This technique can be applied here, too. It makes it possible to semantically link the contents from user feedback to the enriched web content, even when the amount of text being dealt with and the degree of the overlap of the used words and terms in both sources is low. To realise information condensation, only the content that deals with the most important source topics (the most influential topics and their semantic neighbors in form of semantically grouped terms of specific text documents, that strongly affect the meaning of other terms) that have no or only a small overlap with other source topics should be displayed here at first because this content is the most relevant one.

The keyword-based search for content and user feedback is another important functionality to implement content filtering. Using this function, it should also be possible to subscribe to specific topics and to feedback from specific users. Important keywords from the viewed content that can be used to formulate queries for this purpose could be offered instantly. Algorithms such as the graph-based ones for keyword extraction presented in [4], [5] and [6] can be applied on single texts (web pages are a good example) and do not rely on large reference corpora that might be unavailable. Here, classical algorithms for frequency-based keyword extraction like TF-IDF [7] and difference analysis [8] will yield unsatisfying results. The automatically determined keywords...
can then be accepted as query terms by simply clicking on them. This way, a convenient solution to interactively analyse user feedback can be realised.

C. Attaching Web Content

In order to display given user feedback on web documents, this added content could be simply placed at the same position as a user dragged it. However, this simple approach will not be satisfying when dealing with highly dynamic content. Online news magazines constantly present new content under the same URL. Older content will move down and ultimately to subpages. An article will therefore likely not be shown at the same position for a long time. With the approach discussed, user feedback would be shown next to a completely different content. This is undesirable. Therefore, mechanisms need to be developed to link/map the new content from the user feedback to existing web content. For this purpose, HTML formatting instructions such as attributes (e.g., id and name) of DIV elements as well as the textual content inside the involved blocks (innerHTML) can be used to build a strong connection between the respective contents. This way, the structure and the content of web documents is employed to identify and confirm this connection with a high probability. Therefore, these additional metadata need to be stored along with the user feedback provided. They will be used to match currently viewed web content with stored user feedback that is possibly available for it. Due to the consideration of both content and structure for this mapping, even plain text documents can be enriched with matching external content from user feedback when its associated text sections are being displayed by the web browser.

III. TECHNICAL IMPLEMENTATION

In this section, the technical implementation of the system is outlined, whereby the focus is on the realised web browser extension’s use cases and user interface. The extension has been implemented for the web browsers Firefox and Chrome.

A. The “Content Mover” Web Browser Extension

The main functionality of the web browser extension “Content Mover” is to enable users to select any available content on displayed web documents and to drag and drop it to other web documents. Thereby, as users should not have to switch to the destination instantly after they selected specific content, they can collect web content in a toolbar while browsing and drop it to any web documents in doing so. This toolbar can be hidden or shown when needed. Screenshots of the steps on how the user can select interesting web content and collect it using the toolbar is shown in Fig. 2, 3 and 4.
toolbar to a chosen web document and dropped at a specific content. This is shown in Fig. 5 and 6. After performing this drag-and-drop operation, the newly added content is still available in the toolbar ready to be attached to other web documents. The content manager is thereby bound to a specific user account in which also user details can be edited after successful login. Fig. 8 presents the user account’s options.

Also, the newly added content is then available to all users of the browser extension and will be displayed, when the web document is retrieved. Furthermore, the user has the possibility to manage content that has been attached to web documents. For this purpose, a content manager has been added to the extension. This tool enables users to inspect their previous activities and delete content when it is necessary. The client-sided implementation of the web browser extensions has been carried out using several web technologies. HTML, CSS and XUL have been used to realise the GUI and browser integration. The drag-and-drop functionality and application logic have been implemented using JavaScript and Ajax.

**B. The Server Implementation**

The browser extension communicates with a server in order to capture and store the discussed user feedback and to create and change user accounts. For this purpose, the server provides several data access objects (DAOs) to manipulate the different types of data (“UserDao” for personal user data, “DataNodeDao” for “moved” content by a user and “DataToWebDao” for the position of web content) that need to be managed and stored in the database. The server offers in the “Service”-class several endpoints that can be called by clients using different URLs in order to perform operation such as the request of user feedback to a specific web document. In this first and basic implementation, however, the desired semantic mapping between original web content and content obtained from user feedback is not realised, yet. Therefore, the position of the added user content is stored so that the client can display it at the right place. The semantic interpretation and further utilization of the user feedback for content mapping purposes and for workflow generation will be discussed in further contributions. Also, the search functionality for content and user feedback mentioned in the section two is not yet available in the first realisation.

Now, an example request to the service “/getdatatoweb” is given, that returns available user feedback to a specific web document. If the service is running on the local machine, it can be called at the following URL:

http://127.0.0.1:8080/ContentMoverService/getdatatoweb

The service expects the POST-parameters “session” and “url” to be set in the request data and in JSON-format:

```json
request
{
"session": "iu2lcul0cg5reu2um95gvfdh5b",
"url": "http://www.mozilla.org/en-US/about/"
}
```
Additionally, this method can be a basis to recommend related validity of existing links found in web documents. This approach, it is even possible to assess the relevance and "leftTagLocation", "topTagLocation", "htmlData", "url" and "userId" in order to make the client know how the content in "htmlData" is to be positioned, where it actually stemmed from and who provided the user feedback.

This information of origin is especially useful when content dependencies between the user feedback and web documents in question need to be derived automatically in order to evaluate the feedback or to recommend relevant content to the users. For this purpose, algorithms are needed that can even identify the topics of short texts in satisfying quality. The graph-based algorithm presented in [3] extends the HITS algorithm [9] and can determine the main and source topics of texts with a high quality even when the textual basis is sparse. This is achieved by considering the semantic relations between the terms in the text during the analysis. It is now employed in a new algorithm to evaluate the quality of the user feedback provided by estimating its relatedness to the web content and to automatically induce links between web documents and web contents based on content analysis. Using this approach, it is even possible to assess the relevance and validity of existing links found in web documents. Additionally, this method can be a basis to recommend related content to users during browsing sessions, too.

IV. LINK ANALYSIS AND INDUCTION

Usually, only webmasters can put links on their web pages in order to refer users to related web content. As mentioned in the introduction, users on the other hand have only a limited number of options to add own links to those pages, e.g. in form of comments to blog entries. The herein introduced solutions to add web content from different sources to web documents does not only present ways to extend their contents, they establish links between the added and existing content as their web addresses are captured, too. These links gained through user feedback are valuable as they complement existing links. However, for special purposes such as automatic content filtering, the validity of these links needs to be confirmed or denied using content analysis. Hence, not only the content fragments from the user feedback are available for this analysis, also their source documents can be used for this purpose. Therefore, in this section, the authors introduce an algorithm to validate those links based on a previously developed clustering technique for main and source topic detection that builds upon the HITS algorithm [9] to determine the relative importance of nodes in web graphs.

A. Topic Detection using Extended HITS

In [3], a method has been presented by the authors to obtain directed co-occurrence graphs based on the notion of term associations, which will be applied here, too. First, it was shown that the association Assn of term A with term B can be calculated using the following formula derived from the field of association rule mining:

\[
Assn(A \rightarrow B) = \frac{|A \cap B|}{|A|} \cdot \frac{|A|}{|n_{\text{max}}|}
\]  

(1)

Hereby, \(|A \cap B|\) is the number of times term A and B co-occurred in the text on sentence level, \(|A|\) is the number of sentences term A occurred in and \(|n_{\text{max}}|\) is the maximum number of sentences any term has occurred in. This way, a relation of term A with term B with a high association strength can be interpreted as a recommendation of A for B. Also, it was explained that it is sensible to only take into account the dominant association (the one with the greater value of term, when applying this formula for both association directions of the two involved terms). Relations gained by this means are more specific than undirected term relations because of their direction. They resemble a hyperlink on a website to another one. In this case however, it has not been manually and explicitly set and it carries an additional weight that indicates the strength of the term association. The set of all such relations obtained from a text represents a directed co-occurrence graph. This technique is applied to each document that needs to be analysed.

Second, the well-known HITS algorithm to evaluate the relative importance of nodes in web graphs was extended and applied on the co-occurrence graphs gained this way in order to not only determine keywords, but to also detect the source topics in texts. These terms strongly influence the main topics in texts but are not necessarily important keywords...
themselves. The extended HITS algorithm takes into account the strength of the term association provided by the edge weights of the directed co-occurrence graphs. It determines two ordered lists of terms: the authority list that contains the most important keywords and the hub list that comprises the source topics. The update rules of the HITS algorithm to iteratively calculate the authority score \( a(x) \) and hub score \( h(x) \) of a node (term) \( x \) are given by the following formulas:

\[
a(x) = \sum_{v \rightarrow x} h(v) \cdot \text{Assn}(v \rightarrow x) \tag{2}
\]

\[
h(x) = \sum_{x \rightarrow w} a(w) \cdot \text{Assn}(x \rightarrow w) \tag{3}
\]

These rules will be executed until convergence is reached (the calculated values do not change significantly in two consecutive iterations) or a fixed number of iterations has been performed.

These authority and hub lists can now be used to automatically generate topically induced document links. An algorithm for doing so will be introduced in the next subsection.

B. Linking Documents

In order to enhance document retrieval methods, Kurland and Lee [10] used language models to generate asymmetric links between initially returned search results and applied the PageRank [11] algorithm on the obtained document graphs to rerank the results. Although automatically induced links can be noisier than manually created ones [12], this structural reranking was able to improve the quality of the result lists consistently. While Kurland and Lee used the generation probability between two documents to establish those links, in this paper, the authors provide a new algorithm for automatic link induction by measuring the degree of the semantic dependency between two documents in question. Moreover, these links offer the possibility to evaluate the validity of manually created links until a certain degree.

The following steps are necessary to determine such links between documents based on their main and source topics:

---

**Algorithm 1: Generation of topically induced links between text documents**

**Input:** A text corpus  
**Output:** Topically induced links between the documents

1. Remove stopwords and apply a stemming algorithm on all terms in the texts. (Optional)
2. Determine the association values for all pairs of co-occurring word forms using formula 1 for each text \( v \) and use the set of all these relations as a directed co-occurrence graph \( C_v \).
3. Determine for each node \( x \) in \( C_v \) the authority value \( a(x) \) and the hub value \( h(x) \) using the formulas 2 and 3 until convergence is reached (the calculated values do not change significantly in two consecutive iterations) or a fixed number of iterations has been executed. Store the (as a suggestion) 20 most significant authority and hub terms of each document in its authority list \( L_A \) and hub list \( L_H \).
4. Calculate the similarity \( S_{AB} \) between \( L_A \) of a document \( A \) and \( L_H \) of another document \( B \) for all documents pairs, e.g. by determining the overlap of these lists.
5. Establish a link between each text \( A \) to another text \( B \) if the similarity \( S_{AB} \) is greater or equal than a preset threshold and save the similarity value \( S_{AB} \).

---

The intuition behind this approach is that a link from document \( A \) to \( B \) can be established when \( A \)'s most important terms (keywords, main topics) can also be found in \( B \)'s list of source topics. In other words: if \( A \) primarily deals with topics that significantly influence \( B \)'s content, then a topically induced link from \( A \) to \( B \) should be added or it can be confirmed that an existing link from \( A \) to \( B \) is valid from a semantic point of view.

C. Discussion

The algorithm presented in the previous subsection can identify the topical basics of documents and can generate links between documents that show strong topical dependencies, not necessarily similarities. Using this approach, the textual user feedback obtained from the utilization of the introduced web browser extension “Content Mover” can also be analysed and topically grouped in order to realise the needed information condensation as explained in section two. The most important main and source topics and links to added contents and their web addresses could be presented when a user first visits a web document to which added content is available. This will help users to maintain an overview of the original and added content and supports their evaluation of its relevance.

Moreover, in order to automatically generate workflows, in the future, this technique to induce links between topically depending documents and contents can become useful, too. The reason for this is that with this algorithm, it is possible to construct chains of documents that semantically rely on each other as it is the case for the steps in workflows to be carried out. In the process of executing a workflow, each step has a specific context (e.g. location, time, input and output parameters) and usually relies on the results of the previous steps. The detection of semantic dependencies is therefore an important step for automatic workflow generation. In that regard, the user feedback already delivers a valuable preselection of possibly related contents to be analysed.

Found semantic dependencies between documents can be used to recommend the respective other document to users in order to explore the topical basics in detail or to get a broader overview of a topic of interest in a more general document, too. This approach can be especially helpful when it is needed to track a topic to its roots, e.g. in the course of a research to be conducted. This way, users do not even have to enter query terms in order to express their information needs for this purpose as these manually selected query terms might be not appropriate at all for the task at hand. Using the presented approach, users just need to follow links to the suggested related contents. Therefore, this goes beyond a simple search.
for similar documents as it offers a new way to search for related documents and to find background information on a topic of interest. This functionality can also be seen as a useful addition to Google Scholar (http://scholar.google.com/), which offers the possibility to search for similar scientific articles.

V. CONCLUSION

Herein, a system for context-based rearrangement of web content and an associated new algorithm for automatic link analysis and link induction between topically related web documents have been introduced. Possible fields of application of these solutions have been discussed in detail as well. Most importantly, they present a first step towards automatic generation of workflows, an important virtual good in the future, based on user feedback acquired during web browsing sessions and content analysis. As this is a work in progress, there are a number of ways to enhance them.

In order to realise a more accurate mapping between web contents provided by users and the enriched web content in the introduced web browser extension “Content Mover”, the analysis of their structural and textual information is necessary. This way, also highly dynamic web documents can be enriched with the correct web content provided by the user feedback. Also, the extension will be enhanced so that users are actively supported to construct specific workflows by combining existing web contents from different sources and to distribute them. These options will be examined in future contributions and empirical evaluations of the presented solutions will be given, too.

ACKNOWLEDGMENT

The authors would like to thank Prof. Dr. Thomas Böhme of the Institute of Mathematics at the Ilmenau University of Technology for his valuable directions and advices regarding the technical realisation of the presented solutions.

REFERENCES