University of Koblenz-Landau

WikiTax - A tool for taxonomy extraction based on the Wikipedia Category Graph

Dominik Mosen (dmosen@uni-koblenz.de)

April 27, 2013

1 Introduction

Wikipedia - as a collaboratively created encyclopedia - uses several means to organise its information. Examples include links from articles to other related articles forming an article graph [ZG07], stand-alone lists[1] listing articles that describe a certain theme, or portals[2] which are meant to introduce users to the main topics of Wikipedia. Besides these methods, categories[3] can be used to classify articles. Therefore, categories are defined that again contain other categories and articles thus shaping a category graph [ZG07].

While the main purpose of the mentioned means is to provide users with additional or related information to certain topics, the given data can also be used to derive domain specific taxonomies which can be leveraged in several areas, such as in the 101companies project [FLSV12]. This project provides knowledge concerning software technologies, software languages and technological spaces. As it is wiki-based and has similar requirements for structuring its information as Wikipedia, it should be examined how Wikipedia handles the problem of organising its content. In this context, Wikipedia's knowledge on computer science topics could be helpful to improve the 101companies' wiki[4]. At the present time, efforts are made to incorporate the Semantic MediaWiki extension[5] into the 101companies' wiki, which makes MediaWiki[6] contents computer-processable by adding semantic annotations. To develop a meaningful structure, condensed information from Wikipedia on relevant topics could serve as a starting point or reveal missing or under-represented subjects.

[1] `http://en.wikipedia.org/wiki/Wikipedia:Manual_of_Style/Stand-alone_lists`, last visit 17th April 2013

[2] `http://en.wikipedia.org/wiki/Wikipedia:Portal`, last visit 17th April 2013

[3] `http://en.wikipedia.org/wiki/Wikipedia:FAQ/Categories`, last visit 17th April 2013

[4] `http://101companies.org/`, last visit 17th April 2013

[5] `http://semantic-mediawiki.org/`, last visit 17th April 2013

[6] `http://www.mediawiki.org/wiki/MediaWiki`, last visit 17th April 2013

Therefore, this paper describes the tool *WikiTax*, which supports the creation of taxonomies based on the Wikipedia category graph and helps to analyse the structure of Wikipedia for desired domains. For this purpose, the structure the Wikipedia Category Graph is described in §2, before §3 explains the developed tool and gives operating instructions. §4 summarises this paper and gives a short outlook.

2 The Wikipedia Category Graph

Categories in Wikipedia are hierarchically ordered. More precisely, the hierarchy of categories can be considered as a graph consisting of vertices representing the categories and of directed edges indicating a subcategory relationship. Although Wikipedia does not enforce the hierarchical ordering of categories, this holds true for most of them. According to [ZG07], the category graph of the German Wikipedia from May 15, 2006 had 7 backlinks causing cycles (i.e. links pointing from a hierarchically lower category to a higher one and thus invalidating the hierarchical order). Although not explicitly mentioned, the dump reports on Wikipedia[7] indicate that cycles are not by intention. Ignoring these rare backlinks, the category graph can be transformed into a directed acyclic graph (DAG), which enables a hierarchical representation.

3 WikiTax

WikiTax utilises the existing Wikipedia category graph to derive taxonomies. The following sections describe the basic data model of *WikiTax* and explain the general workflow when using it. As the main idea of this workflow is to reduce the Wikipedia category graph until a meaningful structure remains, there are several mechanisms to support this reduction process, which is discussed in section 3.3. Section 3.4 treats the possibilities of reviewing and reverting decisions during the reduction, while the last section 3.5 deals with saving, loading and exporting results and the data formats provided for this purpose.

3.1 The data model

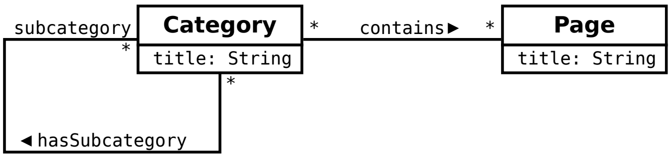


Figure 1: Schema of the Wikipedia category graph

As mentioned before, *WikiTax* works with the Wikipedia category graph. Figure 1 depicts the data model which serves as a basis for *WikiTax*. It consists of `Category`s and

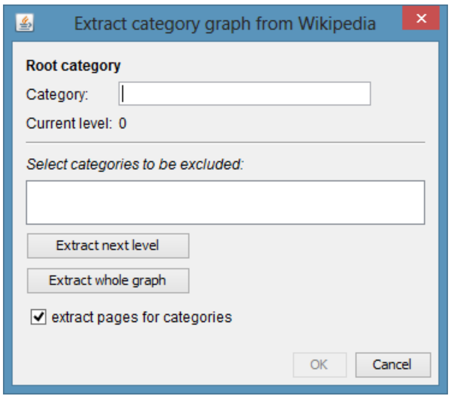
[7] `http://en.wikipedia.org/wiki/Wikipedia:Dump_reports`, last visit 17th April 2013

`Pages` which are connected by `hasSubcategory`- or `contains`-associations. `Categorys` correspond to Wikipedia categories and `Pages` to Wikipedia articles. The `hasSubcategory`-association indicates a subcategory relationship between two Wikipedia categories whereas the `contains`-association describes that a Wikipedia article is contained in a Wikipedia category.

3.2 The Workflow

The derivation of a taxonomy using *WikiTax* is essentially done in two steps. At first, a category subgraph - starting at a given root category - is extracted from Wikipedia. Afterwards, this graph is incrementally reduced until a meaningful structure remains. The user is guided through these steps by two separated views that will be discussed in the following sections.

3.2.1 Extracting the category subgraph


Extract category graph from Wikipedia
Root category
Category:
Select categories to be excluded:
Extract next level
Extract whole graph
OK
Cancel

Current level: 0

☑ extract pages for categories

Figure 2: The extraction dialog

To extract a category subgraph, the extraction dialog pictured in Figure 2 is opened by selecting `File > Extract` from the menu bar after *WikiTax* was started. The user only needs to enter a valid Wikipedia category which is defined as the root category for the extraction process. After the `Extract whole graph` or `Extract next level` button has been clicked, the extraction starts. In both cases, the Wikipedia category graph is traversed in breadth-first search order. This is done by following all `containsPage`- and `hasSubcategory`-edges, starting at the given root category. The traversal is executed by running successive queries against Wikipedia via the MediaWiki API[8].

In the first case, the complete subgraph is traversed and thereby extracted in one run whereas in the second case, only one level is processed at a time. The latter procedure permits to exclude individual categories from the extracted graph at a particular level and to continue the extraction for the next level. This is especially helpful if it is known

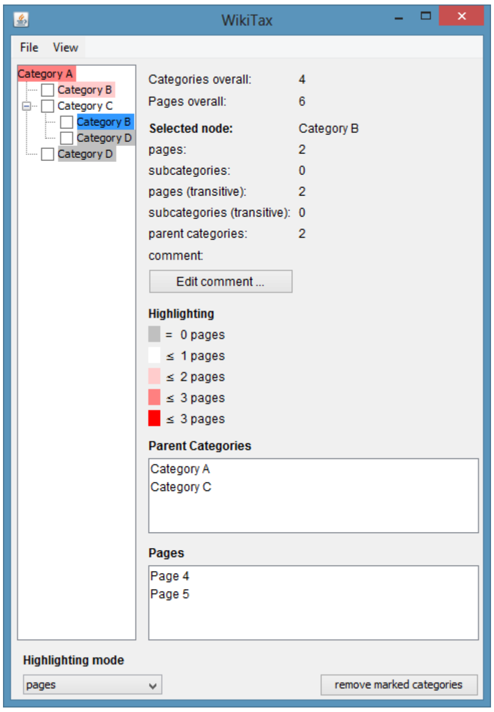
[8] `http://en.wikipedia.org/w/api.php`, last visit 17th April 2013

that the category in question is unnecessary and relatively comprehensive, and would therefore consume a lot of time when extracting.[9]

No matter which extraction option was selected, the progress will be displayed in a progress dialog. The extraction can be cancelled at any time by clicking the `Cancel` button, but every progress since the last extraction step will be lost.

If a satisfying category subgraph was extracted, the extraction dialog is accepted by clicking the `OK` button. After an optional saving of the extracted graph, the 'graph reduction step' begins.

3.2.2 Reducing the graph



Categories overall: 4
Pages overall: 6
Selected node: Category B
pages: 2
subcategories: 0
pages (transitive): 2
subcategories (transitive): 0
parent categories: 2
comment:

Highlighting
▨ = 0 pages
□ ≤ 1 pages
🟥 ≤ 2 pages
🟥 ≤ 3 pages
🟥 ≤ 3 pages

Parent Categories
Category A
Category C

Pages
Page 4
Page 5

Highlighting mode
pages

Figure 3: The reduction view

Although dealing with a graph, the reduction view (figure 3) shows it as a (hierarchical) tree structure. This tree is produced by traversing the graph in depth-first search order. Thereby, cycles are detected and the responsible edges (backward arcs) are blacklisted and ignored. This produces a directed acyclic graph (DAG). Every edge of this DAG corresponds to a tree node in the tree view displayed in the reduction view. As the root category does not have an incoming edge in the DAG (because it was the first category to be extracted), no correspondent edge can be given. Nevertheless the root category

[9] Be aware that - on extraction - the exclusion of categories is purely intended to reduce time effort and complexity of the extracted graph. If you are unsure about exclusion of categories, keep them as they can still be excluded in the reduction step (cf. section 3.2.2).

is still part of the category tree. The relationships between a category graph and its appropriate category tree can be seen in figure 4.

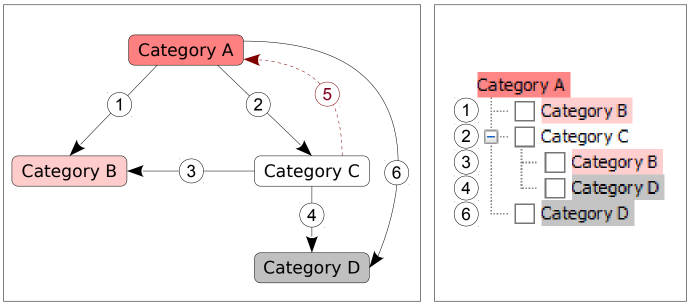


Figure 4: Correspondence between a category graph and a category tree. As `Category A` is the root category it has no corresponding edge. Notice that edge number 5 has no corresponding tree node because it causes a cycle and is therefore not part of the DAG.

The reduction is done by marking nodes through clicking check boxes in the reduction view and by removing them with the activation of the `remove blacklisted categories` button. Internally, the removal is done by blacklisting edges corresponding to the tree nodes in the underlying category graph. The influence of removing a tree node is depicted in figure 5. Moreover, every tree node (except for the root node) can be provided with a comment that is meant to indicate the reason for the exclusion.

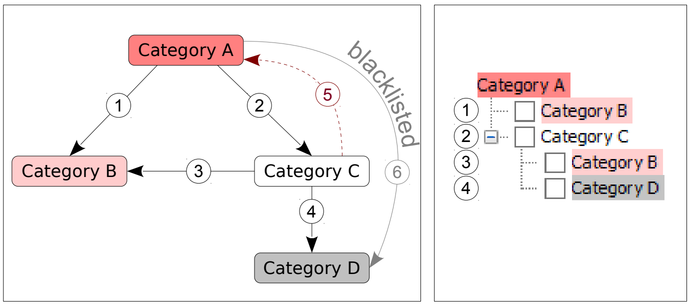


Figure 5: Influence of a removal of a tree node to the graph

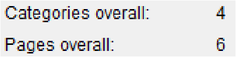
After several iterations of removing tree nodes, a reasonable structure that resembles a domain specific taxonomy should evolve. If it does not, this could be due to a badly organised structure of Wikipedia or an inadequate reduction of categories by the user. However, if an incomplete or unsatisfying structure is obtained, it may serve as a starting point for designing one.

3.3 Reduction process support

The reduction process described in the preceding section is highly dependent on the user's knowledge which enables him or her to include or exclude categories. *WikiTax* has different means to support users in this decision process: that is providing general data about the graph, data for the currently selected tree node and a highlighting system giving visual information which categories could be important or not.

3.3.1 General data

The general data (figure 6) contain information that is not specific for a certain tree node but represent information concerning the whole underlying category graph (i.e. the data can be displayed even if no node is selected):



Categories overall:	4
Pages overall:	6

Figure 6: General data

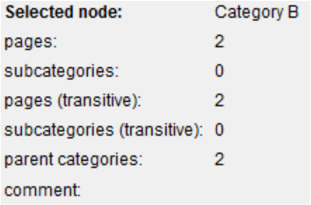
Categories overall The number of different categories the category graph contains overall.

Pages overall The number of different pages the category graph contains overall.

Notice, even if the tree view has several occurrences of the same category or page - based on the graph -, they are only counted once.

3.3.2 Data for a selected tree node

As previously mentioned, every tree node refers to an edge pointing to a category in the category graph. When a tree node is selected, the following data regarding the targeted category are displayed (figure 7):



Selected node:	Category B
pages:	2
subcategories:	0
pages (transitive):	2
subcategories (transitive):	0
parent categories:	2
comment:	

Figure 7: Selected node data

pages The number of pages directly contained in the selected category

categories The number of subcategories directly contained in the selected category

pages (transitive) The number of pages directly or indirectly contained in the selected category

categories (transitive) The number of subcategories directly or indirectly contained in the selected category

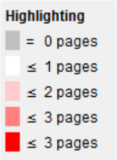
parent categories The number of categories this category is contained in

It holds especially for `pages (transitive)` and `categories (transitive)` that the calculation is based on the graph. Hence, they represent the number of different pages or categories that are reachable from the selected category in the category graph. This means that no category or page is counted twice.

3.3.3 The highlighting system

The highlighting modes listed below can be selected in the highlighting combo box at the lower left of the reduction view:

- pages
- categories
- pagesTransitive
- categoriesTransitive



Highlighting
= 0 pages
≤ 1 pages
≤ 2 pages
≤ 3 pages
≤ 3 pages

Figure 8: Legend for the highlighting mode `pages`

For the highlighting of the `pages` (figure 8), thresholds are calculated to colour the tree nodes. Therefore, the first quartile, the median, the third quartile and the maximum are calculated based on the count of pages directly contained in a category while ignoring categories containing zero pages. Based on these thresholds, the classification shown in table 1 is applied to the tree nodes to determine their colourings. Therefore, the conditions are checked top-down. The first valid condition determines the colouring. The colouring of the `categories` mode is equally defined but instead of the contained pages count, the contained categories count is used.

If highlighting was applied to transitive values, quartiles would provide no suitable visualisation because it is known that tree nodes deeper in the tree will contain less transitive pages and categories than their parent nodes and so their colouring would rather reveal their depths in the tree than their importance. To produce a meaningful highlighting for transitive values, the colouring of every tree node is based on its immediate parent tree node. As before, the colouring is determined by applying the first valid condition displayed in table 2.

Colouring	Condition
gray	pages = 0
white	pages $\leq$ 1. quartile
light red	pages $\leq$ median
medium red	pages $\leq$ 3. quartile
dark red	pages $\leq$ maximum

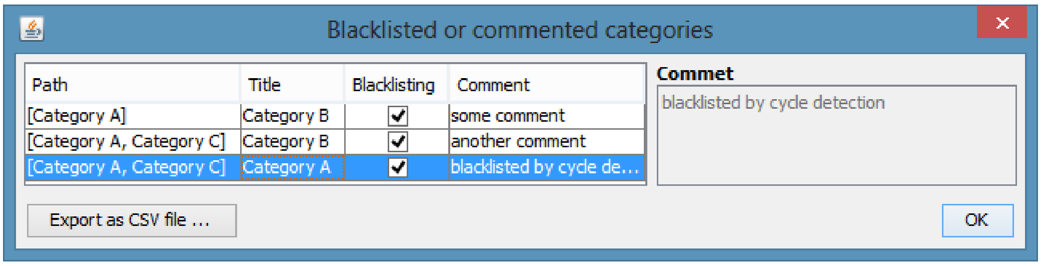
Table 1: Highlighting of tree nodes for mode `pages`

Colouring	Condition
gray	pages (transitive) = 0
white	pages (transitive) $\leq$ 25% of the parent's pages (transitive)
light red	pages (transitive) $\leq$ 50% of the parent's pages (transitive)
medium red	pages (transitive) $\leq$ 75% of the parent's pages (transitive)
dark red	pages (transitive) $\leq$ 100% of the parent's pages (transitive)

Table 2: Highlighting of tree nodes for mode `pagesTransitive`

3.4 Review processed categories

To review all tree nodes that have been blacklisted or provided with a comment, a table dialog (Figure 9) can be opened from the menu bar in the tree view by selecting `View > Table`. The table dialog presents an overview that allows to alter all comments and blacklistings except the ones for the root node and the nodes assigned to a backward edge. To unambiguously identify every tree node, their corresponding tree path is provided. Additionally, the table can be exported and saved as a CSV file [Sha05].



Blacklisted or commented categories

Path	Title	Blacklisting	Comment
[Category A]	Category B	☑	some comment
[Category A, Category C]	Category B	☑	another comment
[Category A, Category C]	Category A	☑	blacklisted by cycle de...

Commet: blacklisted by cycle detection

Figure 9: The table dialog

When clicking the `OK` button, the table dialog is closed and all changes are applied to the underlying tree view.

3.5 Saving, loading and exporting results

The File menu contains three more functions: Save, Load, and Export as JSON. With Save, the underlying graph can be saved while with Load, a saved graph can be restored. As *WikiTax* is based on JGraLab[10], its dedicated TG format for serialising graphs is used to save and load (processed) category graphs. All relevant data for reconstructing the category tree - including the currently calculated statistics for each category node - are stored in the TG file. Furthermore, the category graph can be exported in form of a JSON file [Cro06] that consists of serialised vertices and edges.

```
[
  {
    "id":1,
    "title":"Category A",
    ...
    "type":"Category"
  },
  {
    "start":1,
    "type":"Contains",
    "end":2
  },
  {
    "id":2,
    "title":"Page 1",
    "type":"Page"
  },
  {
    "start":1,
    ...
    "type":"HasSubcategory",
    "end":3
  },
  {
    "id":3,
    "title":"Category B",
    ...
    "type":"Category"
  }
  ...
]
```

Listing 1: Serialised graph in JSON format (shortened)

Listing 1 shows a serialised JSON graph, which consists of five elements, forming a graph composed of three vertices and two edges. It describes a graph consisting of the category named Category A which is connected to the page named Page 1 via a Contains-edge and which is also connected to the category named Category B via a HasSubcategory-edge. Every vertex has an id that the edges use to indicate their start and end. Moreover, each vertex is either a Category or a Page and each edge is either a ContainsPage- or a HasSubcategory-edge. Every vertex can have a title that is its corresponding Wikipedia title. Some other attributes that were omitted in the listing can be obtained from figure 10.

[10] https://github.com/jgralab, last visit 17th April 2013

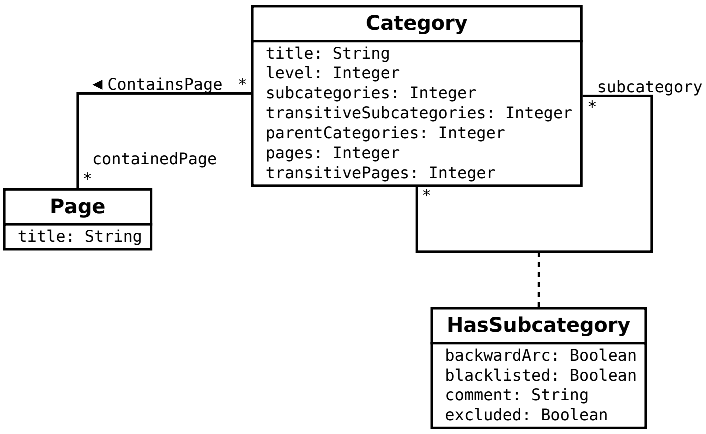


Figure 10: Schema of the *WikiTax* category graph

4 Conclusion

The developed tool *WikiTax* provides basic functions to extract parts of the Wikipedia category graph and to reduce it iteratively in order to carve out the essential structure of Wikipedia. The concepts that are required for this process are explained in an application-oriented manner.

WikiTax should be regarded as a prototypical implementation with the aim to visualise, analyse and extract relevant parts of the Wikipedia category graph. It was therefore generically designed and tested. Hence, it is essential to apply *WikiTax* with a specific target in mind, rather than to perform generic tests while developing it. A need for adjustments of the highlighting modes or of the provided statistics could thus arise to improve the usability and suitability of the tool.

References

[Cro06] Douglas Crockford. The application/json media type for javascript object notation (json). RFC 4627, IETF, 7 2006.

[FLSV12] Jean-Marie Favre, Ralf Lämmel, Thomas Schmorleiz, and Andrei Varanovich. *101companies*: a community project on software technologies and software languages. In *Proceedings of TOOLS 2012*, LNCS. Springer, 2012. 16 pages. To appear.

[Sha05] Y. Shafranovich. Common format and mime type for comma-separated values (csv) files. RFC 4180, IETF, 10 2005.

[ZG07] Torsten Zesch and Iryna Gurevych. Analysis of the wikipedia category graph for nlp applications. In *Proceedings of the TextGraphs-2 Workshop (NAACL-HLT 2007)*, pages 1–8, January 2007.