Rainer Kuhlen (Hrsg.)

Information: Droge, Ware oder Commons?

Wertschöpfungs- und Transformationsprozesse auf den Informationsmärkten

Proceedings des 11. Internationalen Symposiums für Informationswissenschaft (ISI 2009)

Konstanz, 1.—3. April 2009

vwh
Verlag Werner Hülsbusch
Fachverlag für Medientechnik und -wirtschaft

Mapping Bibliographic Records with Bibliographic Hash Keys

Jakob Voß[1], *Andreas Hotho*[2], *and Robert Jäschke*[2]

[1] Verbundzentrale des GBV (VZG), Gottingen
[2] Knowledge & Data Engineering Group, University Kassel

Abstract

This poster presents a hash key for bibliographic records called bibkey. It is shown how bibkey can be used to detect duplicates and to map similar bibliographic records among distributed databases.

1 Introduction

To manually seek a specific publication, one can start with known metadata fields (title, author, ...) and use experience and background knowledge until you localize it. In the same way – despite the vast heterogeneity of citation styles and metadata formats – it is relatively easy to find out whether two citations or bibliographic records refer to the same publication. But computer programs need unique identifiers or intelligent heuristics to point to a publication or to detect whether two records are duplicates. Normal publication identifers are assigned centralized either by publishers (ISBN, DOI, ...) or by bibliographing institutions (OCLC number, LCCN number, . . .). Bibkey is a simple approach to create a hash key for bibliographic records that can be calculated by anyone who knows the author (or editor), title, and year of a publication. The goal is to support the search process by pointing the user to similar references.

2 Specification and Implementation

The bibliographic hash key is calculated based on four metadata fields: title, author (or editor if there is no author), and year. The fields are normalized and concatenated in a defined way to form *bibkey level 0*:[1]

1. Fields are normalized by Unicode case folding to NFKC lowercase.
2. All characters but digits (year), and Unicode letters (title), and dot or whitespace (author) are removed, whitespaces become one space.
3. The author field is split into names by the string `'and'`.
4. Names are normalized, de-duplicated, sorted, and joined by `','`
5. The final string is: `'title [names] year'`.

Bibkey level 0 can be used for string comparisons and to form more elaborated keys. In particular *bibkey level 1* is generated by calculating the MD5 checksum and prepending the digit `'1'`. The hardest part of implementation turned out to be full Unicode support for NFKC lowercase, letters, spaces, and sorting. Reference implementations and test cases are available in Perl and Java as well as a public web form. The following example contains a bibliographic record and its bibkeys:

Author:	Trudi Bellardo Hahn and Charles P. Bourne
Title:	A History of Online Information Services, 1963-1976
Year:	2003

Level 0:	`ahistoryofonlineinformationservicesl9631976`
	`[t.hahn,c.bourne] 2003`

Level 1:	`123d1561cl9c8546d292e4a9e1eafflfO`

3 Related Work

Bibliographic identifiers were discussed and developed especially in the late 1990s. Most identifiers cannot be derived from existing metadata. The Serial Item and Contribution Identifier (SICI) is a rarely used exception that relies on very clean metadata. The query string of an OpenURL can also be seen as a complex identifier to point to a specific publication. Many methods of du-

[1] See details at http://www.gbv.de/wikis/cls/Bibliographic_Hash_Key.

plicate detection calculate keys, signatures, or fingerprints for each record to reduce the number of comparisons. Such keys are also used in digital libraries to detect duplicates and in several implementations of FRBR work detection (OCLC, VCOB, Virtua, ...). Bibkey is created similarly ad-hoc from basic metadata (title, author, year). Without having to refer to any authority or a complicated data format it maps each unique record to one simple hash.

4 Usage, Status, and Outlook

Bibkey level 1 was first used as *interhash* by the social cataloging application BibSonomy [1] to detect if the same publications have been entered by different users.[2] Other applications (for instance the Kölner Universitäts-Gesamtkatalog, KUG) can quickly look up via Bibkey whether a publication already exists in BibSonomy. Currently Bibkey is formalized as standard and analyzed in strength and limitations. Thereby two kinds of error exist: first, same publications could be mapped to different keys and second, different publications could be mapped to one key. It turned out that the first error depends on the quality of the metadata and the definition of “same publication” and the second error only occurs in special cases like anonymous works or works without known year and articles with standard titles like “Introduction”, “Book Reviews” or “News”. Further development of bibkey will aim on reducing errors of the first kind by removing diacritics and using only part of a title and on testing the benefit of bibkey for FRBR work detection.

References

[1] A. Hotho/R. Jäschke/C. Schmitz/G. Stumme. BibSonomy: A social bookmark and publication sharing system. In *Proceedings of the CS-TIW*, pages 87–102, Aalborg, Denmark, 2006.

[2] http://bibsonomy.blogspot.com/2007/11/detecting-duplicates-in-bibsonomy.html

GBV VZG

Verbundzentrale des GBV

ENDOWED CHAIR OF THE HERTIE FOUNDATION
Knowledge and Data Engineering
ELECTRICAL ENGINEERING & COMPUTER SCIENCE, UNIVERSITY OF KASSEL

Bibliographic Hash Keys

Mapping Bibliographic Records

Jakob Voß[1], Andreas Hotho[2], Robert Jäschke[2]

This poster presents a set of hash keys for bibliographic records called bibkeys. Unlike other methods of duplicate detection, bibkeys can directly be calculated from a set of basic metadata fields (title, authors/editors, year). It is shown how bibkeys are specified and used to map similar bibliographic records in BibSonomy, among distributed library catalogs, and other distributed databases.

Motivation

To check whether two citations or bibliographic records refer to the same publication, either manual work or unique identifiers or good heuristics of duplicate detection are needed. Centralized identifiers (ISBN, DOI, LCCN etc.) cannot be derived from other metadata fields but must be looked up. Other systems like OpenURL [1] and SICI [2] require detailed and clean metadata which you rarely find in normal citations. Methods of duplicate detection are common in digital libraries but they mostly build on direct comparisons of full records or multiple comparisons of minimal distances of multiple signatures [3]. Methods or FRBR work detection use similar methods or they are bound to specific bibliographic record formats or authority files [4]. In Contrast bibkeys can be applied by anyone who knows the authors (or editors), title, and year of a publication. An important feature of bibkey is that records are matched without having to directly compare them. Instead a bibkey is calculated by a simple method for each record and can directly be matched .

Examples

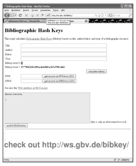
Given the book with authors "Trudi Bellardo Hahn and Charles P. Bourne", title „"A History of Online Information Services, 1963-1976", published in 2003, these metadata fields are joined to a string (bibkey level 0) and its checksum to bibkey level 1:

bibkey level 0: ahistoryofonlineinformationservices19631976 [t.hahn,c.bourne] 2003

bibkey level 1: 14ed100f75dd4459cffeb272bdbc2d1e7

Author names are abbreviated by splitting the names into tokens at white spaces. If the first and the last token are equal, this is returned once as surname. Otherwise the the first character of the first token (given name) followed by a dot is prefixed to the last token (surname). Some examples of both cases:

"knuth knuth"	"knuth"
"knuth"	"knuth"
"donald e. knuth"	"d.knuth"
"d.e. knuth"	"d.knuth"
"donald knuth"	"d.knuth"



Bibliographic Hash Keys

This script calculates Bibliographic Hash Keys (bibkey) based on title, author/editor, and year of a bibliographic resource.

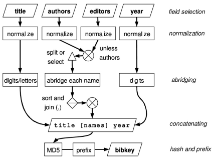
Title:
Author:
Editor:
Year:
Bibkey level 0: []

calculate bibkey
PPN: get record via PPN from GSO
ISBN: get record via ISBN from GSO
See also the Web interface at BibSonomy.
(this is only an abbreviated record)
post to BibSonomy

check out http://ws.gbv.de/bibkey/

Specification

A general bibkey is based on four metadata fields: title, authors, editors, and year. The editors field is only used if no authors are given. First all fields are normalized. The authors/editors field is either split into single names or the first author is selected. Names are abridged, sorted and joined to a comma-separated list. Year and title are reduced to digits or digits and letters. Finally the fields are concatenated as title + " [" + names + "] " + year. Either this string is used or the MD5 Message-Digest Algorithm checksum [5] of its UTF-8 representation plus a prefix makes a hashed bibkey.



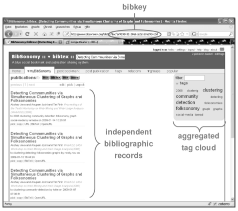
For a specific bibkey version, normalization, abridging, sorting, concatenating, and the prefix need to be defined. Bibkey level 1 uses Unicode Normalization Form Compatibility Composition (NFKC) [6], case folding to lower case, and replacement of white spaces with one space, except at the beginning and and of a string as normalization. All author names are used and the abridging method is shown in the examples-section above. The reduction to letters and digits in the title respects all Unicode letter. The prefix that is added to a MD5 checksum is "1", so in total a bibkey has 33 characters from a z, 0 9. The latest specification is available online as well as reference implementations in Java and Perl.

Specification
http://www.gbv.de/wikis/cls/Bibliographic_Hash_Key

Reference implementations
http://ws.gbv.de/bibkey/
http://www.bibsonomy.org/help/doc/inside.html

Usage

Bibkey version level 1 is used as "interhash" by the social bookmarking application BibSonomy [7] to detect if the same publication has been entered by different users.



BibSonomy::bibtex::{Detecting Commmunities via Simultaneous Clustering of Graphs and Folksonomies} - Mozilla Firefox

http://www.bibsonomy.org/bibtex/1acfec953843b168e61e2e167e29b4c3d

BibSonomy :: bibtex :: Detecting Commmunities via Simu

A blue social bookmark and publication sharing system.

Detecting Commmunities via Simultaneous Clustering of Graphs and Folksonomies
Akshay Java and Anupam Joshi and Tim Finin *Proceedings of the Tenth Workshop on Web Mining and Web Usage Analysis (WebKDD)* (2008)
to 2008 clustering community detection folksonomy graph social-media by anneba on 2009-01-14 10:25:57

Detecting Commmunities via Simultaneous Clustering of Graphs and Folksonomies
Akshay Java and Anupam Joshi and Tim Finin *WebKDD 2008 Workshop on Web Mining and Web Usage Analysis* (2008)
to clustering detecting folksonomies graphs by nesty.nov on 2009-01-10 16:44:24

Detecting Commmunities via Simultaneous Clustering of Graphs and Folksonomies
Akshay Java and Anupam Joshi and Tim Finin *WebKDD 2008 Workshop on Web Mining and Web Usage Analysis* (2008)
to clustering community detection by folke on 2009-01-07 07:06:51

bibkey
logged in as hotho · settings · logout · help · blog · about
Home
▾myBibSonomy
post bookmark
post publication
tags
relations
▾groups
popular
publications (5)
RSS BibTeX RDF more
previous | 1 | next
edit | pick | unpick
pick | copy | URL | BibTeX | OpenURL
filter:
tags
2008 clusterig clustering community detecting detection folksonomies folksonomy graph graphs social-media toread
independent bibliographic records
aggregated tag cloud
Fertig
zotero

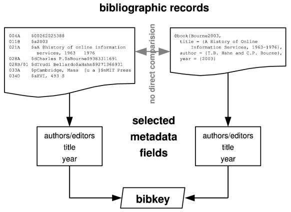
The bag-model of social tagging allows each user to manage its own bibliographic records that then can be aggregated. Other applications can quickly look up by bibkey, whether a given publication has already been entered in BibSonomy. For this purpose BibSonomy provides a JSON API and VZG provides a wrapper to the SeeAlso Linkserver protocol [8]. The Kölner Universitäts Gesamtkatalog (KUG) indexes its records with bibkey and uses it to link BibSonomy. Lookup of records via bibkey in other library catalogs and in the Wikipedia project is planned.

Usage of BibSonomy JSON API
http://www.bibsonomy.org/help/addons/integration

SeeAlso Services at VZG
http://ws.gbv.de/seealso/services/

Use of bibkey in OpenBib/KUG
http://blog.openbib.org/2008/07/15/neue-version-von-kug-und-openbib/

Planned bibliographic record store for Wikipedia using bibkeys
http://de.wikipedia.org/wiki/Benutzer:Duesentrieb/Biblio



bibliographic records

```
004A    $00262025388
011@    $a2003
021A    $aA @history of online information
          services, 1963   1976
028A    $dCharles P.$aBourne$9383311691
028B/01 $dTrudi Bellardo$aHahn$9271366931
033A    $pCambridge, Mass  [u a ]$nMIT Press
034D    $aXVI, 493 S
```

```
@book{Bourne2003,
  title = {A History of Online
      Information Services, 1963-1976},
  author = {T.B. Hahn and C.P. Bourne},
  year = {2003}
```

Evaluation and Outlook

Each bibkey method defines a binary classifier for duplicate detection of bibliographic records. Thereby two kinds of error exist: first, same publications could be mapped to different keys (false negative) and second, different publications could be mapped to one key (false positive). It turned out that the first error highly depends on quality of the metadata and the definition of "same publication". Sensitivity can further be increased by improvement of the normalization step and by selecting only the first author/editor. A next version of bibkey should for instance normalize by removing all diacritics. Improvement in the abridging step can also help, especially with organizations as authors. Abridging could also include usage of authority files but this would limit the ease of bibkey usage. The second error only occurs in special cases like anonymous works, works without known year and for articles with standard titles like "Introduction", "Book Reviews" or "News" which are frequently found in journals. Further development of bibkeys will aim on testing its benefit for FRBR work detection by removing the year field and on usage of bibkeys as link targets on the Semantic Web.

References

[1] ANSI/NISO. The OpenURL Framework for Context-Sensitive Services. 2004 (Z39.88).

[2] ANSI/NISO. Serial Item and Contribution Identifier. 1996 (Z39.56).

[3] L. Padmasree, V. Ambati, J.A. Chandulal and M.S. Rao. Signature Based Duplication Detection in Digital Libraries. In *Proceedings of the ICUDL,* Alexandria, 2006. http://tera-3.ul.cs.cmu.edu/conference/2006/25.pdf

[4] T. Hickey, J. Toves. FRBR Work-Set Algorithm. OCLC, 2005. http://www.oclc.org/research/software/frbr/frbr_workset_algorithm.pdf

[5] R. Rivest. The MD5 Message-Digest Algorithm. 1992 (RFC 1321).

[6] M. Davis, M. Dürst. Unicode Normalization Forms. Revision 29, 2008-03-28 (Unicode Standard Annex #15). http://www.unicode.org/reports/tr15/tr15-29.html

[7] A. Hotho, R. Jäschke, C. Schmitz, and G. Stumme. BibSonomy: A social bookmark and publication sharing system. In *Proceedings of the CS TIW*, p 87 102, Aalborg, 2006. http://www.kde.cs.uni-kassel.de/stumme/papers/2006/hotho2006bibsonomy.pdf

[8] J. Voß. SeeAlso: A Simple Linkserver Protocol. Ariadne 57, 2008. http://www.ariadne.ac.uk/issue57/voss/

[1] http://www.gbv.de

[2] http://www.kde.cs.uni-kassel.de/