Mapping Bibliographic Records with Bibliographic Hash Keys

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

[1] Verbundzentrale des GBV (VZG), Göttingen
[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*:[3]

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 comparisions 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

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

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: `ahistoryofonlineinformationservices19631976 [t.hahn,c.bourne] 2003`
Level 1: `14ed100f75dd4459cffeb272bdbc2d1e7`

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 duplicate 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.[4] Other applications (for instance the Kölner UniversitätsGesamtkatalog, 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, and G. Stumme. BibSonomy: A social bookmark and publication sharing system. In *Proceedings of the CS-TIW*, pages 87–102, Aalborg, Denmark, 2006.

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