Claude Goutorbe

Document Interlinking in a Digital Math Library

In: Petr Sojka (ed.): Towards a Digital Mathematics Library. Grand Bend, Ontario, Canada, July 8-9th, 2009. Masaryk University Press, Brno, 2009. pp. 85--94.

Persistent URL: http://dml.cz/dmlcz/702560

Document Interlinking in a Digital Math Library

Claude Goutorbe

Cellule Mathdoc, Université Joseph Fourier and Centre National de la recherche Scientifique, Grenoble, France

Abstract. Document interlinking is one of the prominent features that users expect from a digital library. Links may be internal (allowing easy navigation within a given literature repository) or may reference external documents living in other digital repositories, as well as the two main reviewing databases in Mathematics (namely Mathematical Reviews and Zentralblatt-MATH).
We describe the linking system that has been implemented within the NUMDAM project.

1 Introduction

Mathematical works always build on previous results, and are thus part of what can naturally be considered a network of literature. This was the case even in the past, long before the advent of electronic communications. But the digital infrastructures that have been built over the last decades now potentially make this network of knowledge widely and easily available.

Our goal is thus to put a given work into a rich context of related documents:

- errata, other parts of a multipart work
- existing reviews of this work in the main mathematical databases (Mathematical reviews, Zentralblatt-MATH, Jahrbuch). These reviews are of interest not only because they provide an introduction, description, or criticism of a given work, but also because they provide additional information such as mathematical classification, keywords, links to other related items, and often provide a way to acquire or obtain access to the actual text of a document.
- access to works that are cited in the reference section. These works may or may not reside in the same digital repository
- access to works that reference the item

In the NUMDAM digital library, some of these relations are established during the construction of the library's catalogue: they record such facts as:

- a given article is an erratum to another article in the library (and conversely)
- a given article is the sequel of another article in the library

In the following, we do not describe how these relations are established, but focus on links that are created automatically from the catalogue data.

Petr Sojka (editor): DML 2009, Towards a Digital Mathematics Library, pp. 85–94.
 ISBN 978-80-210-4781-5

Good bibliographic databases exist in the field of Mathematics, and can potentially be used to identify a given mathematical work, enabling cross-referencing of items.

These databases should be equipped with software that is able to cope with a number of commonly found problems such as: any given work may be cited in a number of different ways, these citations may contain errors (which may be simply human errors or may have been introduced by the digitization process, e.g. optical character recognition errors).

We describe an identifying tool that is being implemented for the Zentralblatt-MATH database.

2 User requirements

Potential users of such software include:

- Digitization projects or publishers who want to enrich their data with interdocument links
- Mathematicians who want to quickly check a citation or access a given work through a simple cut and paste interface

Different digitization projects and publishers, as well as end users (mathematicians) cannot currently be expected to structure citation data in a uniform way, if at all.

The "lowest common denominator" is in fact raw strings generated by optical character recognition, the publishing system or interactive copy and paste, and the software should be able to use these as input.

3 Techniques for record matching

When trying to decide whether two citation strings actually refer to the same work, two main issues have to be addressed: which data should be used for comparison and under which conditions the data identify the same resource.

The first class of techniques involve a preliminary data preparation phase, during which the records are parsed and individual parts are tagged to conform to some standard representation. These tagged records are then compared field by field, and some criteria are applied to decide the overall results of the comparisons. This approach presents several difficulties in the context of citation matching:

1. While the tagging process does in principle greatly reduce the structural heterogeneity of records, individual fields often remain coded in different ways and cannot be compared using exact comparison methods
2. Errors in the citation strings to be matched often result in a given field comparison to fail entirely
3. The "structuring" process is both costly and error prone

For these reasons, it seems appropriate to develop tools that make no assumptions on the structure of the citation strings and do not attempt to identify *a priori* the individual logical components of a reference string.

The discovery of a citation's structure may actually be the outcome of the matching process.

In the world of mathematics, such a tool already exists for the MathSciNet database, and is used routinely by some digitization projects to generate links to MathSciNet reviews.

In the following pages, we describe some of the difficulties involved in the implementation of such tools, the techniques and heuristics that can be used. We then present the results obtained by software that has been implemented by the NUMDAM project to enable linking to the Zentralblatt-MATH and `minidml` databases.

4 The matching challenge

In the context of this work, the matching problem can be defined as follows:

Given a database of bibliographic items, and a bibliographic reference string, find database entries that describe the same work as the reference string.

Reference strings usually have the following characteristics:

- they are noisy: typing errors, optical recognition errors
- they are innacurate: wrong volume numbers, wrong page numbers, wrong journal titles, wrong publication year, etc...
- they are incomplete: missing authors, no title, incomplete bibliographic data
- titles may be translated from the original
- while being perfectly accurate, their coding might be different from the one used in the database: different transliteration of author names, roman numerals/arabic numerals for volume or part numbers, different paging scheme: some journals have a double paging scheme (per issue, per volume)
- journal titles abbreviations may be just about anything

It is thus clear that when deciding whether a given database item matches the reference string, we will have to use some metrics and/or to apply certain heuristics.

More precisely, the matching process that has been implemented is a sequence of steps reducing the space of near matches, initially defined as the whole bibliographic database, to a small number of items. Each step uses specific metrics or applies a specific set of heuristics.

5 String metrics

Many string similarity metrics have been proposed [1]. Character based metrics such as the Levenshtein distance and others can handle typing or optical

character recognition errors that occur in a given field, but do not work well when used on the complete reference string, because subfields are usually ordered in a different way.

Token based metrics may be used to compensate for this different ordering by matching substrings independently of their position. These substrings (tokens) may be words or substrings of length n, known as n-grams. The use of n-grams has the added advantage of being tolerant to small mistakes and variations in spelling.

The matching software use both kinds of metrics, during different steps.

In the next sections, we first give some statistics about the results that have been obtained, then describe the matching process in a more detailed way, and finally give some examples.

6 Matching results

These are notoriously difficult to evaluate. The following can happen:

1. No match is found:
   - there is actually no match
   - the software does not meet expectations
2. Matches are found but are not relevant

Here are however some figures computed using different datasets.

1. Journal articles from the NUMDAM project: metadata is of good quality (reference strings are accurate). Depending on the journal coverage in Zentralblatt, we get up to 96 % of exact matches
2. Bibliographic references cited by these same articles: metadata may be noisy because of optical recognition errors and inaccurate or incomplete because of authors' mistakes. They include every possible kind of reference (journal articles, books, thesis, reports,... The average rate of matches is 75 % of the total number of bibliographic items, and may grow up to 85 %, depending on the journal. A large part of these matches has been checked during the development of the software, meaning that these figures include a very low rate of irrelevant matches
3. Bibliographic references from the *Journal of Differential Geometry* (project Euclid). The matching rate is 89 %, including dubious or irrelevant matches (no checking was performed)

7 The matching strategy

The overall strategy that is used is the following:

1. Generate possible candidates.
2. Rank candidates by evaluating their similarity to the reference to match.
3. If no good candidate is found, possibly restart at step 1.
4. Depending on the number of plausible matches that are required, output the first n candidates. For example, when using an interactive tool, the user might be interested in seeing multiple results.

7.1 Generating possible candidates

Since it is not feasible (in terms of computing cost) to compare a reference string with every database record, we need a way to select database records that are potential matches.

This amounts to running a boolean query on the target database. For this purpose, the target database has been equipped with an index on the words occurring in a bibliographic citation.

During this phase, we have to decide which parts of the reference string are to be used in the initial query.

The intuition that numbers (volume, issue, pages, publication year) are of particular interest is supported by some simple statistics. The following figures have been generated by scanning part of the Zentralblatt database, collecting author names, volume number, publication year and paging information for each entry.

Total number of journal articles: 413721
v = volume number
y = publication year
fp = first page number
lp = last page number
t = first (significant) title word
a = first author name (without initials)
Total number of different v | fp-lp strings: 376038 (90.89)
Total number of different a | y | fp strings: 402594 (97.31)
Total number of different t | fp-lp strings: 406844 (97.92)
Total number of different a | fp-lp strings: 406844 (98.33)
Total number of different a | v | fp strings: 410735 (99.28)
Total number of different a | y | fp-lp strings: 411959 (99.57)
Total number of different a | v | y | fp strings: 412350 (99.67)
Total number of different a | v | fp-lp strings: 412710 (99.76)
Total number of different a | v | y | fp-lp strings: 412889 (99.80)

These figures show that, in conjunction with author names, numbers occuring in the bibliographic data almost identify a given journal article.

This leads to the first strategy used in this step: *use numbers if possible*.

Since author names usually occur near the beginning of the reference string, a query using one of the first few words (i.e. the first word that actually occurs in the database index) in conjunction with numbers should quickly pinpoint a few entries of interest. If there is no author name in the reference string, the first few words are probably title words, and the above table shows that the same property holds.

This strategy may fail in the presence of wrong data, or be unusable because the reference string contains no numbers. In this case the initial query is formulated using the first tokens of the reference string.

7.2 Ranking candidates

Having obtained a set of potential matches, we have to use some kind of similarity metrics to evaluate the quality of a match.

For each candidate, we first compute the *cosine similarity* with the given reference string, using n-grams vectors. This allows in particular for small mistakes and variations in spelling. n-grams are a set of n consecutive characters from the input string. After some experimentation, it appears that a value of $n = 3$ is a reasonable choice.

It turns out that this first step usually reduces the candidate set to a very small size, often to the point that it contains only one database item.

At this point, we have a set of *structured* database items, and we may use other metrics to further rank this set.

- Approximate substring matching (similar to *agrep*) is used to check author names.
- The similarity of numbers is computed using the Dice coefficient.
- Paging information is matched. Page numbers that differ only by one are considered equal.
- Titles can potentially be discovered and compared using approximate substring matching.

There is actually no need to perform all these steps: some simple heuristics allow the process to be stopped early.

8 Matching in action

In this section, we show the matching process in action and give some examples of rules that are used when deciding if a given match is "good enough".

8.1 Using numbers

An easy case Consider the following reference:

```
Grimeisen G.,  Gefilterte Summation von Filtern und iterierte
Grenzprozesse. I, Math. Ann., 141 (1960) 318-342.
```

The initial query looks up "Grimeisen" and yields a set R of 31 database items. The set S of numbers in the reference is (141, 1960, 318, 342). It turns out that only one item in R also carries these four numbers

```
Zbl 0096.26201
Grimeisen, Gerhard
Gefilterte Summation von Filtern und iterierte Grenzprozesse. I.
Math. Ann. 141, 318-342 (1960).
```

with a cosine similarity of 0.96. This is considered a good match, and the process goes no further (note that the reference string is a very accurate one).

Using page numbers The following example illustrates the fact that the string *author + first page + last page* identifies a given item in most cases. The reference

```
P G. KONTOROVIC - K.M. Kutyev,  On the theory of lattice-ordered
 groups (in Russian),
 Izv. Vys. Uč. Zav. Mat., 1959, 112-120
```

is mapped to

```
Zbl 0093.24701
Kontorovic, P.G.; Kutyev, K.M. Zur Theorie der verbandsgeordneten
 Gruppen. Izv. Vyssh. Uchebn. Zaved., Mat. 1959, No.3(10),
 112-120 (1959).
```

even though the article's titles are written in different languages.

Numbers and the Dice coefficient The Dice coefficient is a similarity measure for sets X and Y, defined as

$$d = \frac{2\,|X \cap Y|}{|X| + |Y|}$$

In the absence of an obvious paging string, we use this coefficient to evaluate the similarity between number sets. The threshold used depends on the (already computed) cosine similarity: a high value of the Dice coefficient is required when the cosine similarity is low, a lower value is acceptable when the cosine similarity is high.

The exact values have been tuned during software testing.

The following example is almost a limit case, with a Dice coefficient of $2.\frac{4}{14} = 0.57$, and a cosine similarity of 0.48.

The reference

```
[2] N. V. BANITCHOUK,  V. M. PETROV,  F. L. TCHERNOUSSKO,
  Résolution numérique de probl\‘emes aux limites variationnels
 par la méthode des variations locales.
 Journal de Calcul numérique et de Physique mathématique,
 tome 6, n. 6, Moscou, 1966, pages 947 \‘a 961.
```

is mapped to the following database item

```
0.488957436865 Banichuk, N.V.; Petrov, V.M.; Chernous'ko, F.L.
 The solution of variational and boundary value problems by
 the method of local variations.
 U.S.S.R. Comput. Math. Math. Phys. 6, No.6, 1-21 (1966);
 translation from Zh. Vychisl. Mat. Mat. Fiz. 6, 947-961 (1966).
```

8.2 What value is a good cosine similarity?

The candidate set should not be reduced too much when computing the cosine similarity: even candidates with a low score have to be considered: the cosine value may be low due to

- titles in different languages (see the preceding section)
- no title in the reference string

```
[6] J. C. SLATER et J. G. KIRKWOOD, Phys. Rev., 37, 1931, 682.

0.647039633316 Slater, J. C.; Kirkwood, J. G.
 The van der Waals forces in gases. Physical Review (2) 37,
 682-697 (1931).
```

- no author names in the reference string
- bibliographic data is very incomplete, especially for books where the bibliographic data in the database is very complete, while the reference is much less verbose

We have found empirically that a value of 0.4 is a good cutting threshold.

Conversely, a high cosine similarity is not always sufficient to declare the database item to be a good match. This is typically the case when the citation string refers to a work that is not referenced in the database, while a work by the same author(s) and a similar title *is* referenced (one could argue that this is sometimes a good match). We have found that a unique match with cosine similarity $\geqslant 0.9$ is almost always the right one, stopping the matching process.

8.3 When numbers cannot be used

This is the case when:

- the reference string has no numbers
- the previous strategy of using numbers has failed, because there are "too many" wrong numbers (wrong paging string, low Dice coefficient)

In this case, we try to match each part of the reference string (e.g. authors, title, bibliographic data).

- Author names are matched in the same way as above, using approximate substring matching.
- Titles are matched in the same way. A number of cases may occur, depending on the database item's title alignment within the reference string; if successful, this step locates the start of the bibliographic data in the reference string.
- Depending on the document type of the database item under consideration, different metrics are used to evaluate the similarity of bibliographic data.

Journal articles Article references usually include an abbreviated journal title. There is no point in trying to lookup an abbreviation in some preconstructed table, because the creativity of authors seems to be unbounded. If X is the set of (abbreviated) words in the reference string, Y is the set of (abbreviated) words in the database item's journal abbreviation, we define $|X \cap Y|$ to be the number of common prefixes (excluding certain common words) and compute the Dice coefficient under this definition. For example, for the two strings

```
J. Diff. Geom.
Jour. of Differ. Geometry
```

the common prefixes are "J", "Diff" and "Geom", giving a Dice coefficient of $2.\frac{3}{3+3} = 1$

Books and monographs Book references have characteristics that makes their matching a challenge of its own. The full bibliographic data recorded in the database includes a wealth of information: collection title, edition statement, place of publication, publisher, publication year, while reference strings use only a small subset of these. Authors include only *important* information which is sufficient to describe the book, mainly: publisher and publication year, sometimes publication place. These strings are usually not abbreviated and appear in the same form in the reference string and in the database (modulo errors !), even if in a different order.

Hence a token based metric is probably a good choice.

The Dice coefficient and similar metrics that are a measure of the number of common tokens are not sufficient and do not distinguish between important and "verbose" information.

We have decided to use a metric based on *inverse database frequency*, similar to what is widely known as *TFIDF* in information retrieval [2].

The set of common tokens is generated, and a score is computed as the sum of weights (inverse database frequencies) of terms in this set. Such a metric gives a higher weight to less frequent terms and appears to work well in practice, even if further investigations are needed to provide hard evidence of that claim.

It does not, however, allow an easy distinction between different editions or printing of a given work (except when the publisher is different for example). Publication years can be used for this, since they can be extracted from the reference string with high confidence, using a simple regular expression. We have found publication years to be innacurate in a surprising number of cases, and allow a gap of one year when comparing them.

9 Conclusion

We have described methods to map a reference string to a citation in a bibliographic database. These methods require only a very shallow analysis of the input string, and rely on a number of string similarity evaluation methods,

as well as some ad hoc heuristics that work reasonably well in practice in the context of mathematical databases.

The software is mainly used in a batch processing environment, where a large number of references, as produced for example by a digitising project, need to be matched without human supervision.

It can be easily adapted to an interactive environment, where a mathematician can check his citations and sometimes discover similar works, such as alternative versions, translations, or multiple parts of a given document.

References

1. William W. Cohen, Pradeep Ravikumar, Stephen E. Fienberg: A Comparison of String Distance Metrics for Matching Names and Records, available at `http://www.cs.cmu.edu/~wcohen/postscript/kdd-2003-match-ws.pdf`
2. S.E. Robertson, K. Spärck Jones: Simple, proven approaches to text retrieval, available at `http://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-356.pdf`