

BMatch: a Semantically Context-based Tool Enhanced by an Indexing Structure to Accelerate Schema Matching

Fabien Duchateau, Zohra Bellahsène and Mathieu Roche

Laboratoire d'Informatique, de Robotique et de
Microélectronique de Montpellier
Université Montpellier II, France

BDA'07
Marseille, France

Table of Content

- 1 Introduction
 - Introduction
 - Contributions
- 2 BMatch Approach
 - Quality Aspect
 - Performance Aspect
- 3 Related Work
- 4 Experiments
 - Quality Aspect
 - Performance Aspect
- 5 Conclusion and Future Work

Introduction

- Discovering semantic correspondences between 2 schemas still a challenging issue
- Semi automatic matchers available based on several approaches [Rahm and Bernstein, 2001, Euzenat et al., 2004]

Motivations

Terminological measures are not sufficient, for example:

- mouse (computer device) and mouse (animal) \Rightarrow polysemia
- university and faculty \Rightarrow totally dissimilar labels

Structural measures have some drawbacks:

- it propagates the benefit of irrelevant discovered matches
- not efficient with small schemas

Performance aspect: handling both numerous and large schemas

Motivating Example

Two heterogeneous schemas from university domain with :

- complex mapping, i.e *courses* with *grad courses* and *undergrad courses*
- not obvious mapping, i.e *staff* with *people*

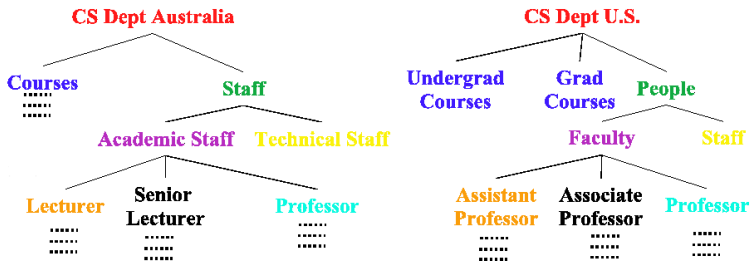


Figure: Mappings discovered by an expert share the same color

Contributions

Our approach: BMatch

BMatch evaluates the similarity between two terms from different schema trees. It has the following properties:

- it is based on **the combination of terminological measures** (Levenhstein and n-grams) **and structural measures** (cosine measure applied to contexts)
- it is both **automatic** and not language-dependent
- it does not use any dictionary or ontology
- it provides an acceptable matching quality
- it relies on an **indexing structure** to accelerate the matching

- 1 Introduction
 - Introduction
 - Contributions
- 2 **BMatch Approach**
 - Quality Aspect
 - Performance Aspect
- 3 Related Work
- 4 Experiments
 - Quality Aspect
 - Performance Aspect
- 5 Conclusion and Future Work

BMatch

BMatch approach consists in two aspects:

- quality aspect, by combining several similarity measures (terminological and structural)
- performance aspect, by enhancing the process with the B-tree indexing structure

Definitions

Context of node n_c

- represents the most important neighbour nodes n_i for n_c
- each neighbour n_i is assigned a weight depending on the relationship n_c

$$\omega(n_c, n_i) = 1 + \frac{K}{\Delta d + |\text{level}(n_c) - \text{level}(n_a)| + |\text{level}(n_i) - \text{level}(n_a)|}$$

Our **String Matching** measure is the average between

- Levenhstein distance
- 3-grams

A Terminological Example



Figure: XML schemas relative to university

- $3\text{grams}(\text{Courses}, \text{GradCourses}) = 0.2$
Courses and GradCourses share 5 common 3grams
- $\text{Lev}(\text{Courses}, \text{GradCourses}) = 0.42$
4 replacements are required to transform the label Courses into GradCourses

⇒ **StringMatching(Courses, GradCourses) = 0.31**

A Context Example

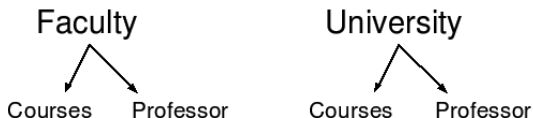


Figure: In the second schema, *Courses* replaces *GradCourses* due to StringMatching value

- $\text{StringMatching}(\text{Faculty}, \text{University}) = 0.002$

- $\text{Context}(\text{Faculty}) = \{\text{Faculty}, \text{Courses}, \text{Professor}\}$

- $\text{Context}(\text{University}) = \{\text{University}, \text{Courses}, \text{Professor}\}$

$\Rightarrow \text{CosineMeasure}(\text{Context}(\text{Faculty}), \text{Context}(\text{University})) = 0.37$

Matching Process

Discovering semantic similarities:

- String Matching between 2 node labels
- if above a given threshold, replacement of one of the label by the other.

Cosine Measure using context:

- due to replacements, the contexts of two nodes can be very similar

Similarity between two nodes

It is the best value between String Matching and Cosine Measure.

Flexibility with the Parameters

- **nb_levels** restricts the context by limiting the number of levels
- **min_weight** restricts the context by keeping only nodes with a weight above this threshold
- **replace_threshold** stands for the minimum StringMatching value to replace one label by the other
- **k** represents the importance given to the context

Flexibility

These parameters allow more flexibility. Tuning them is required in some specific scenarii.

Conclusion about the quality

The combination of terminological measures and the context increases the probability to discover relevant matches.

However, the schemas are parsed twice.

Solution

An indexing structure is used to accelerate the matching process.

The B-tree

The B-tree is an indexing structure to quickly retrieve an element.

- balanced tree (for fast search)
- each node is composed of $M-1$ indexes and has M children nodes

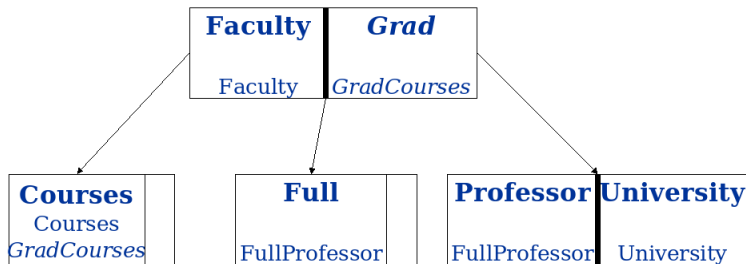


Figure: An example of B-tree

Reducing the Search Space with the B-tree

Assumption: most of the terminologically-close labels share a common token. Thus the B-tree indexing structure enables to restrict the search space.

Algo BMatch :

- labels are divided into tokens
- each token is an index in the B-tree with references to all labels containing this token
- match search of a label is limited to the labels referenced by the common tokens
- else the whole B-tree may be searched using the cosine measure

- 1 Introduction
 - Introduction
 - Contributions
- 2 BMatch Approach
 - Quality Aspect
 - Performance Aspect
- 3 **Related Work**
- 4 Experiments
 - Quality Aspect
 - Performance Aspect
- 5 Conclusion and Future Work

Related Work

COMA++ [Aumueller et al., 2005]

- combination of many terminological measures and a user-defined synonym table
- a matrix is built for each couple of elements and for each measure
- a strategy is applied to select the mappings
- mappings are modified and/or validated by the user

Similarity Flooding [Melnik et al., 2002]

- a simple string matching algorithm to provide initial matchings
- structural rules and propagation to refine the matchings
- mappings are modified and/or validated by the user

- 1 Introduction
 - Introduction
 - Contributions
- 2 BMatch Approach
 - Quality Aspect
 - Performance Aspect
- 3 Related Work
- 4 **Experiments**
 - Quality Aspect
 - Performance Aspect
- 5 Conclusion and Future Work

Experiments

Experiments focuses on two aspects:

- quality aspect, by comparing with another schema matching tool
- performance aspect, by showing the benefit of using the B-tree

Schemas Used for the Quality Experiments

The two schemas from the university domain (2):

- widely used in the literature due to the difficulty to match them
- small schemas
- their labels are not heterogeneous

More experiments with other sets of schemas from various domains (biology, business order, etc) have been performed. The results are satisfying when compared with other matchers (COMA++, Similarity Flooding).

Discovered Mappings by BMatch and COMA++

Element from schema 1	Element from schema 2	Relevance
Professor	Professor	+
CS Dept Australia	People	
Courses	Grad Courses	+
CS Dept Australia	CS Dept U.S.	+
Courses	Undergrad Courses	+
Academic Staff	Faculty	+
Staff	People	+
Technical Staff	Staff	+
Senior Lecturer	Associate Professor	+

Table: BMatch discovered mappings (threshold set to 0.15)

Element from schema 1	Element from schema 2	Relevance
Professor	Professor	+
Technical Staff	Staff	+
CS Dept Australia	CS Dept U.S.	+
Courses	Grad Courses	+
Courses	Undergrad Courses	+

Table: COMA++ discovered mappings

Precision, Recall and F-measure

	Precision	Recall	F-measure
COMA++	1	0.56	0.72
BMatch	0.89	0.89	0.89

Table: Results of COMA++ and BMatch on the XML schemas

Note that BMatch parameters are set to default. An optimal configuration enables to obtain a 1.00 F-measure.

Schemas Used for the Performance Experiments

Schemas come from two sets, OAGIS¹ and XCBL²:

- large number of schemas related to business order
- both small and large schemas
- their labels are normalized
- publicly available

The aim of these performance experiments is to show that the B-tree indexing structure accelerates the matching.

¹www.oagi.org

²www.xcbl.org

The B-tree is Useful when the Number of Schemas Increases

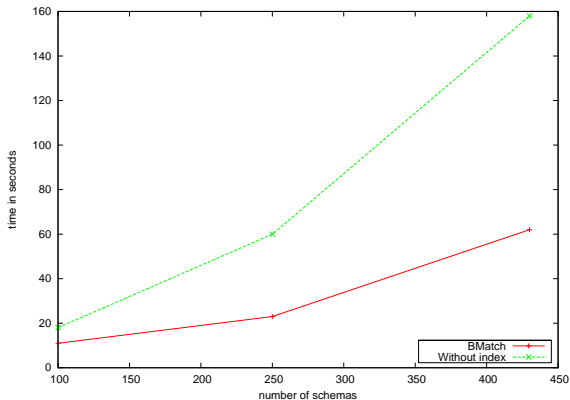


Figure: Comparison of the performance with and without the indexing structure, depending on the number of schemas

The B-tree is Efficient with High Number of Schema Nodes

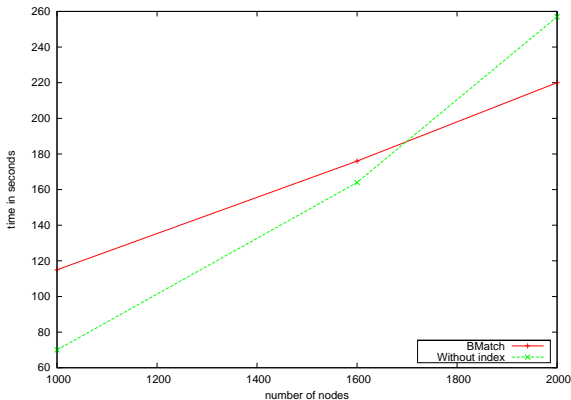


Figure: Comparison of the performance with and without the indexing structure, depending on the size of the schemas

- 1 Introduction
 - Introduction
 - Contributions
- 2 BMatch Approach
 - Quality Aspect
 - Performance Aspect
- 3 Related Work
- 4 Experiments
 - Quality Aspect
 - Performance Aspect
- 5 Conclusion and Future Work

An automatic schema matching approach





- based on the combination of terminological and structural measures
- with an acceptable quality of matching
- handles many large schemas
- flexible thanks to the parameters

However

- tuning is not automatic, but some tools could handle this step (eTuner)
- more heterogeneity in the experiments

Ongoing work

- exploring other index structures (hashtables)
- refining the discovery of complex mappings

-  Aumueller, D., Do, H., Massmann, S., and Rahm, E. (2005).
Schema and ontology matching with coma++.
In SIGMOD 2005.
-  Euzenat, J. et al. (2004).
State of the art on ontology matching.
Technical Report KWEB/2004/D2.2.3/v1.2, Knowledge Web.
-  Melnik, S., Molina, H. G., and Rahm, E. (2002).
Similarity flooding: A versatile graph matching algorithm and its
application to schema matching.
*In Proc. of the International Conference on Data Engineering
(ICDE'02).*
-  Rahm, E. and Bernstein, P. A. (2001).
A survey of approaches to automatic schema matching.
VLDB Journal: Very Large Data Bases, 10(4):334–350.