A Flexible Approach for Planning Schema Matching Algorithms

Fabien Duchateau, Zohra Bellahsène and Rémi Coletta

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

CooPIS'08
Monterrey, Mexico

Table of Content

1. Introduction
   - Introduction
   - Related Work
   - Motivations
   - Contributions
2. MatchPlanner
   - Overview
   - Decision Trees
   - Matching with a decision tree
   - Discussion
3. Experiments
   - Quality Aspect
   - Performance Aspect
4. Conclusion and Future Work

Introduction

Schema Matching

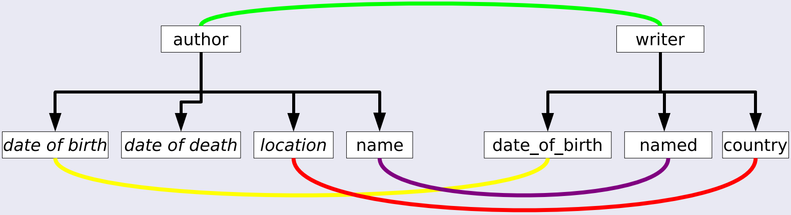


Figure: Discovering semantic correspondences between 2 schemas still a challenging issue in many applications

Semi automatic matchers combine several match algorithms to improve matching quality [Rahm and Bernstein, 2001, Euzenat et al., 2004]

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

Motivations

A brutal aggregation function entails drawbacks:

- **quality** → more weight to closely-related match algorithms can have a negative impact
- **flexibility** → how to aggregate new match algorithms ?
- **threshold** → one threshold for each match algorithm instead of a global one
- **performance** → useless measures are computed.

Recall vs precision:

- most matching tools **promote precision**
- easier to remove irrelevant discovered matches than finding relevant missed matches → **recall seems a better choice**

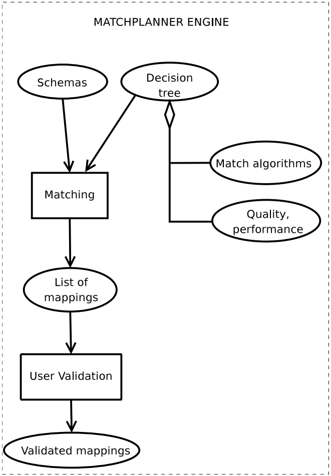
Contributions

Our approach: MatchPlanner:

- it is based on decision trees to combine match algorithms and avoid previous drawbacks.
- notion of planning in the schema matching process.
- a tool has been designed based on the planning approach.
- experiments demonstrate that our tool provides good performance and quality of matches w.r.t. the main matching tools.

1. Introduction
   - Introduction
   - Related Work
   - Motivations
   - Contributions
2. MatchPlanner
   - Overview
   - Decision Trees
   - Matching with a decision tree
   - Discussion
3. Experiments
   - Quality Aspect
   - Performance Aspect
4. Conclusion and Future Work

MatchPlanner



Input: schemas to be matched
a decision tree

Algo: for each pair of schema elements, match it with the decision tree.

Output : list of matches (optionnally validated by an expert)

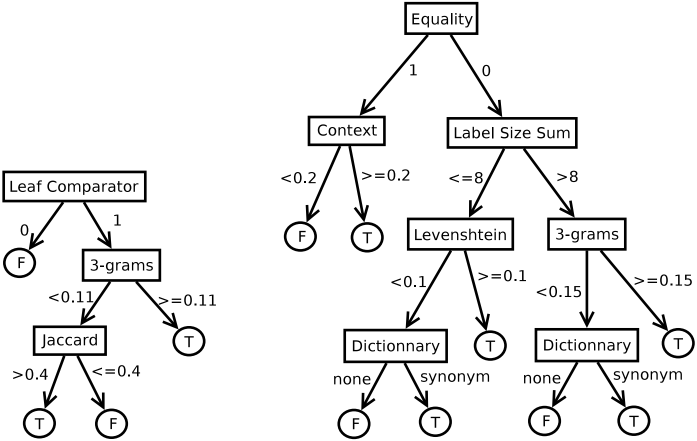


Figure: Examples of decision trees

Definitions

A decision tree contains plans (i.e ordered sequences) of match algorithms. More formally, it is a set of

- internal nodes → the match algorithms
- edges between 2 nodes → conditions on the result of match algorithms
- leaf nodes → the relevance of the match

Features

- **performance**, in terms of discarded match algorithms
- **quality**, minimum F-measure obtained during training phase (for learned decision tres only)

Example of matching with a decision tree

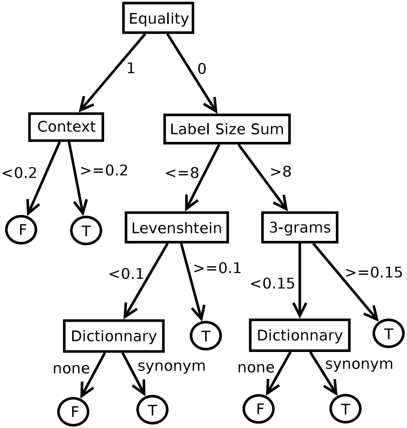
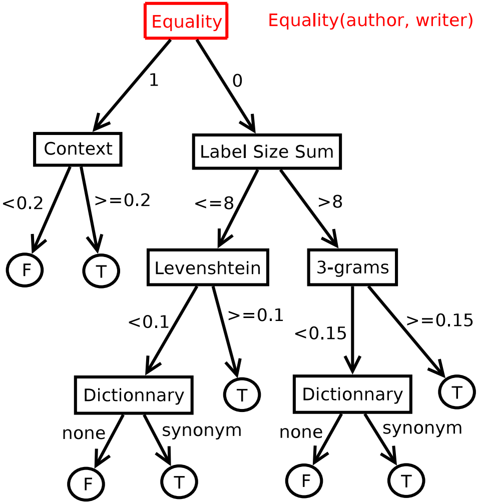
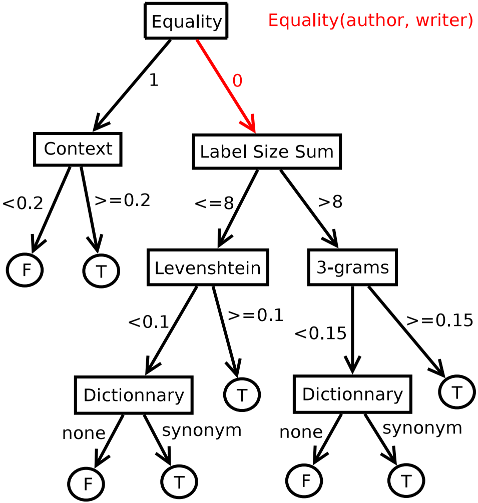


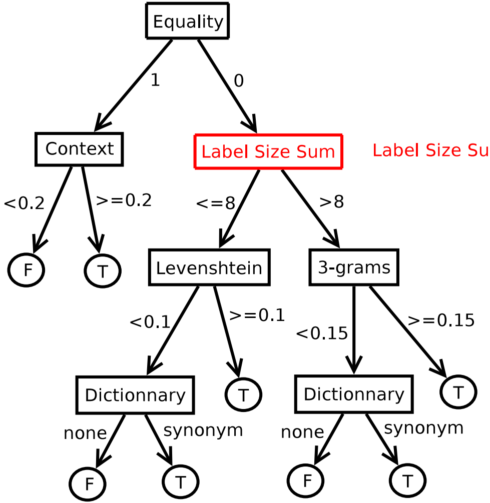
Figure: How to match the pair of elements *(author, writer)* with this decision tree ?



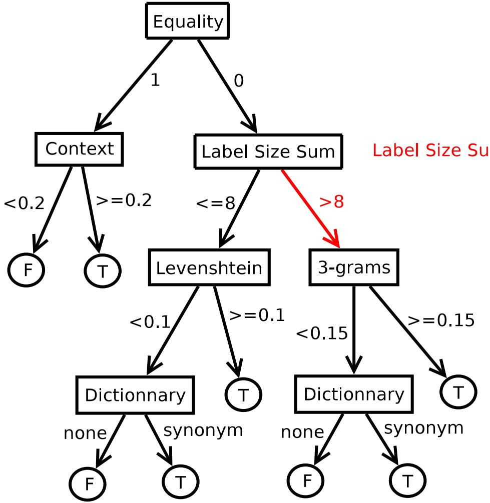
Equality(author, writer) = 0



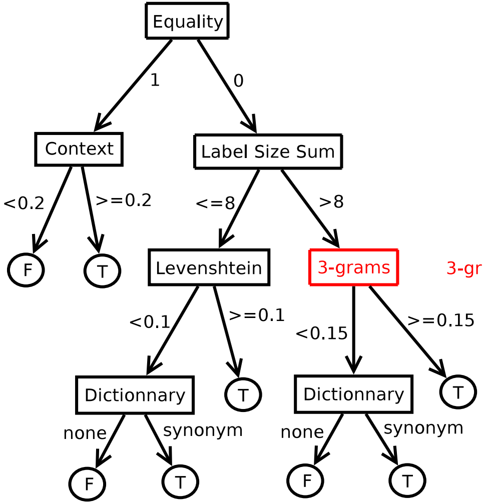
Equality(author, writer) = 0



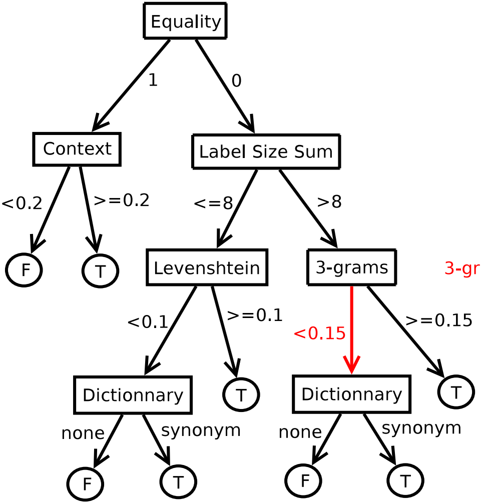
Label Size Sum(author, writer) = 12



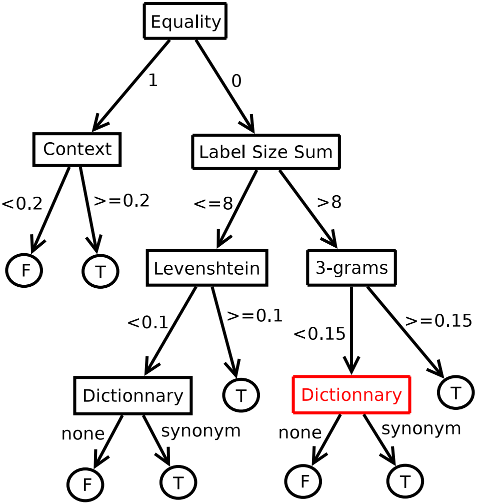
Label Size Sum(author, writer) = 12



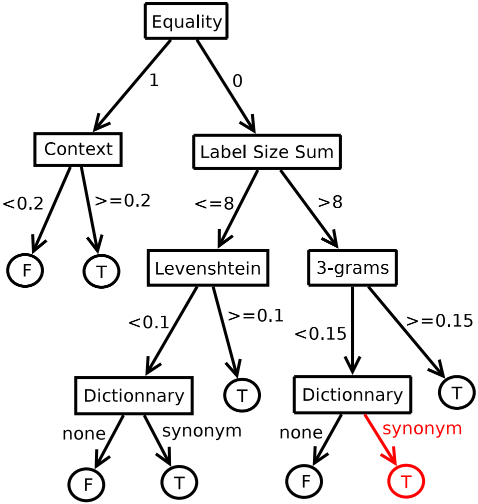
3-grams(author, writer) = 0.125



3-grams(author, writer) = 0.125



Dictionary(author, writer) = synonym



Dictionary(author, writer) = synonym

Discussion

Advantages of the decision trees

- simple to understand or interpret (boolean logic).
- handles both numerical and categorical data.
- many related match algorithms cannot have a very strong impact on a similarity value, thus improving matching quality.
- threshold is specific for each match algorithm.
- applies only a subset of the match algorithms, thus improving performance.

Shortcoming

How to build reliable or appropriate decision trees ?

Discussion

Advantages of the decision trees

- simple to understand or interpret (boolean logic).
- handles both numerical and categorical data.
- many related match algorithms cannot have a very strong impact on a similarity value, thus improving matching quality.
- threshold is specific for each match algorithm.
- applies only a subset of the match algorithms, thus improving performance.

Shortcoming

How to build reliable or appropriate decision trees ?

Discussion

Advantages of the decision trees

- simple to understand or interpret (boolean logic).
- handles both numerical and categorical data.
- many related match algorithms cannot have a very strong impact on a similarity value, thus improving matching quality.
- threshold is specific for each match algorithm.
- applies only a subset of the match algorithms, thus improving performance.

Shortcoming

How to build reliable or appropriate decision trees ?

Discussion

Advantages of the decision trees

- simple to understand or interpret (boolean logic).
- handles both numerical and categorical data.
- many related match algorithms cannot have a very strong impact on a similarity value, thus improving matching quality.
- threshold is specific for each match algorithm.
- applies only a subset of the match algorithms, thus improving performance.

Shortcoming

How to build reliable or appropriate decision trees ?

Discussion

Advantages of the decision trees

- simple to understand or interpret (boolean logic).
- handles both numerical and categorical data.
- many related match algorithms cannot have a very strong impact on a similarity value, thus improving matching quality.
- threshold is specific for each match algorithm.
- applies only a subset of the match algorithms, thus improving performance.

Shortcoming

How to build reliable or appropriate decision trees ?

Discussion

Advantages of the decision trees

- simple to understand or interpret (boolean logic).
- handles both numerical and categorical data.
- many related match algorithms cannot have a very strong impact on a similarity value, thus improving matching quality.
- threshold is specific for each match algorithm.
- applies only a subset of the match algorithms, thus improving performance.

Shortcoming

How to build reliable or appropriate decision trees ?

1. Introduction
   - Introduction
   - Related Work
   - Motivations
   - Contributions
2. MatchPlanner
   - Overview
   - Decision Trees
   - Matching with a decision tree
   - Discussion
3. Experiments
   - Quality Aspect
   - Performance Aspect
4. Conclusion and Future Work

Experiments

Comparison with COMA++ and SF on two aspects:

- quality (precision, recall and F-measure)
- performance (time in seconds)

Seven scenarios:

- **book** and **university** (widely used in the literature)
- **thalia** (benchmark with the courses offered by some American universities)
- **travel** (airfare web forms)
- **person** (describing people)
- **currency** and **sms** (popular web services).

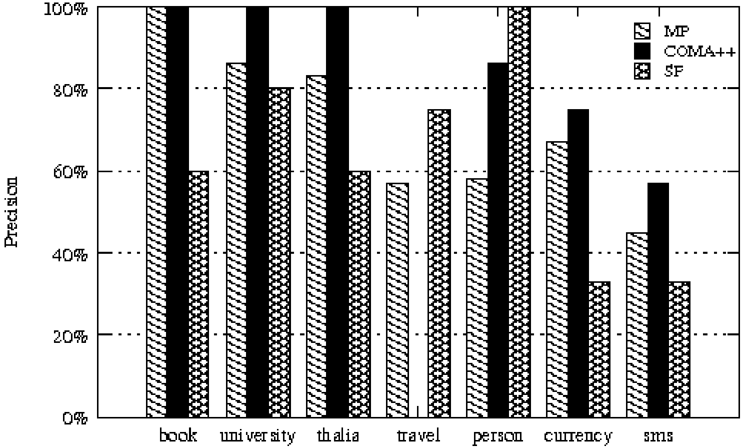


Figure: COMA++ achieves the best precision in 5 scenarios

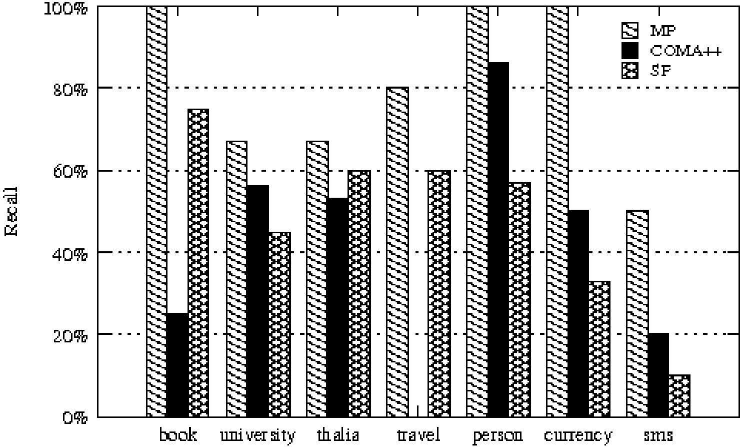


Figure: MP obtains the highest recalls (mostly above 60%) and it discovers all the relevant matches for 3 scenarios

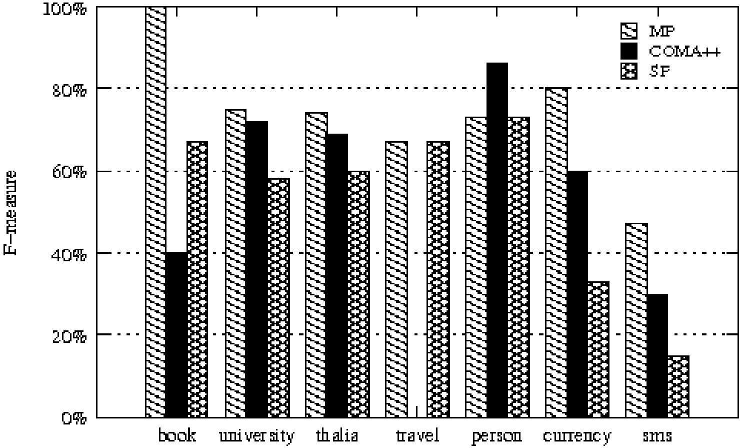


Figure: MP performs best on 6 scenarios

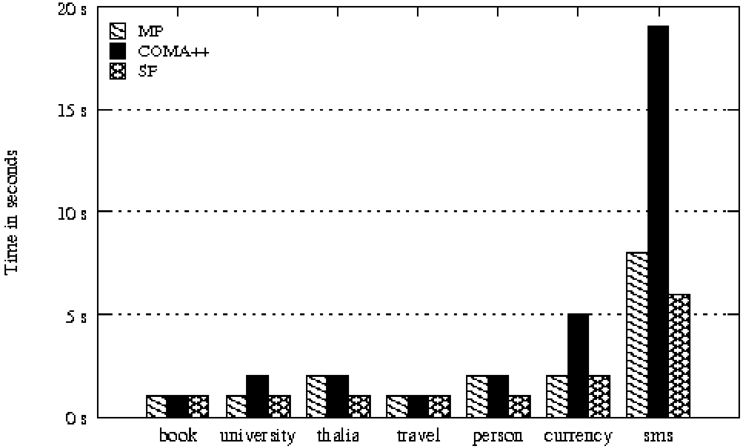


Figure: Time performance for matching each scenario

1 Introduction
- Introduction
- Related Work
- Motivations
- Contributions

2 MatchPlanner
- Overview
- Decision Trees
- Matching with a decision tree
- Discussion

3 Experiments
- Quality Aspect
- Performance Aspect

4 Conclusion and Future Work

MatchPlanner, a new schema matching approach

- based on decision trees to plan match algorithms
- flexible and it promotes recall
- outperforms the existing matching tools on the quality aspect
- provides an acceptable time performance

Ongoing work

- automatic generation of decision trees with machine learning techniques
- improving results with expert feedback
- comparing our approach with SMB ;-)

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.