Analyzing Mapping Extraction Approaches

C. Meilicke, H. Stuckenschmidt

Computer Science Institute
University of Mannheim
A5, 6 68159 Mannheim, Germany
{christian, heiner} @informatik.uni-mannheim.de

Abstract. While lots of research in ontology matching is related to the issue of computing and refining similarity measures, only little attention has been paid to question how to extract the final alignment from a matrix of similarity values. In this paper we present a theoretical framework for describing extraction methods and argue that the quality of the final matching result is highly affected by the extraction method. Therefore, we discuss several extraction methods and apply them to some of the results submitted to the OAEI 2006. The results of our experimental study show that the proposed strategies differ with respect to precision and recall. In particular, theoretical considerations as well as emprirical results indicate that methods that additionally make use of information encoded in the ontologies result in better extractions compared to state of the art approaches.

1 Motivation

Automated matching systems have to be applied to detect semantic relations between ontologies representing overlapping domains. The majority of matching systems approach this problem by computing similarities between the entities (concepts, roles, ect.) to be matched. While similarities can be obtained by comparing the matchable entities one by one, structure and semantic based techniques use more sophisticated ways to compute, propagate and refine similarities by taking the context of these entities into account. As a final step of the matching process an alignment has to be extracted. Since the similarities computed in the first phases of the matching process suggest a diversity of matching hypotheses, the final result will be highly affected by the extraction process. Obviously, it makes sense to use the structural and semantic information in the process of similarity computation. We argue that the same holds for the extraction process. Therefore, we propose a theoretical framework to distinguish between several extraction methods and present as well as evaluate some extraction algorithms that make use of the additional information encoded in the ontologies to be matched.

1.1 Problem Statement

In accordance to Euzenat and Shvaiko [1] the problem of ontology matching can be defined as follows. For each ontology $\mathcal{T}$ there is a function $Q(\mathcal{T})$ that defines matchable elements of $\mathcal{T}$. Given ontologies $\mathcal{T}_1$ and $\mathcal{T}_2$, the task of matching is to determine correspondences between $Q(\mathcal{T}_1)$ and $Q(\mathcal{T}_2)$. Correspondences can be defined as 4-tuples

$\langle e, e', r, v \rangle$ where $e \in Q(\mathcal{T}_1)$ and $e' \in Q'(\mathcal{T}_2)$, r is a semantic relation, and $v \in [0, 1]$ is a confidence value. In this paper, we only consider the simple case where $Q(\mathcal{T})$ are the concepts of $\mathcal{T}$ and r is the equivalence relation. Given correspondence c we use the functions $source(c)$ and $target(c)$ to refer to the aligned concepts, $rel(c)$ to denote the semantic relation, and $conf(c)$ to refer to the confidence value.

The extraction problem we address is a subproblem of the matching problem. A process that solves this problem takes a mapping $\mathcal{M}$ (a set of correspondences) and two ontologies $\mathcal{T}_1$ and $\mathcal{T}_2$ as input and returns a mapping $\mathcal{M}' \subseteq \mathcal{M}$ as result. In many matching systems the final mapping is extracted from a similarity matrix S. Notice that for each similarity matrix S there exists a mapping $\mathcal{M}$, such that each correspondence in $\mathcal{M}$ corresponds to a cell in S. The methods we propose to solve this problem will thus be relevant for the large class of matching systems that first compute similarity matrices or comprehensive intermediary mappings and thereafter extract alignments as final outcome. An optimal solution to an extraction problem is a mapping $\mathcal{M}' \subseteq \mathcal{M}$ such that each $c \in \mathcal{M}'$ truly reflects the semantic relation $rel(c)$ between $source(c)$ and $target(c)$, and there exists no $\bar{c} \in \mathcal{M} \setminus \mathcal{M}'$ such that $\bar{c}$ truly reflects the semantic relation $rel(\bar{c})$ between $source(\bar{c})$ and $target(\bar{c})$.

1.2 Related Work

The extraction problem has only been of minor interest in research related to ontology matching. Euzenat and Shvaiko [1] spend five pages on the topic in their comprehensive book 'Ontology Matching'. They mainly discuss thresholds and a greedy startegy as well as several methods that optimize the results of a one-to-one extraction. Some of these approaches are taken up and extended in the following sections. In opposite to our approach Euzenat and Shvaiko model the extraction problem to be independent of the ontologies to be matched. To our suprise, the papers submitted to the OAEI 2006 contest describing the matching systems of the participants contain only minor information on extraction. We believe that the importance of the extraction problem is highly underestimated in research and that extraction methods have strong effects on characteristics and quality of the resulting mappings.

In prior work [7] we applied the notion of mapping consistency to eliminate potentially erroneous correspondences from automatically generated mappings. Not being aware of the applicability to the more general problem of mapping extraction we referred to this process as repairing of mappings. This approach has been extended and modified towards an extraction technique in [6] where it has been evaluated on synthetic data sets. The main contribution of these approaches compared to state-of-the-art extraction methods is based on the fact that knowledge endcoded in the ontologies cannot only be used for computing similarities but also plays a crucial role in the final extraction process.

1.3 Outline and Contribution

In section 2 we introduce the notion of an extraction function and describe several distinctive properties of extraction functions. In particular, we distinguish between ontology dependent functions that make use of the additional knowledge encoded in the ontologies and ontology independent functions. In section 3 we take up this distinction and present several algorithmic implementations of both ontology independent (section 3.1) and ontology dependent extraction functions (section 3.2). We implemented these algorithms and conducted several experiments on some of the results submitted to the OAEI 2006. The experimental results (section 4) indicate that certain extraction methods, in particular ontology dependent methods, yield better solutions compared to naive approaches. Nevertheless, our main contribution is to emphasise the importance of the extraction problem and to point out that different extraction techniques have strong effects on the characteristics and the quality of the final matching result.

2 Preliminaries

We understand the extraction process as the application of a sequence of extraction functions to an input mapping and the ontologies to be matched. Therefore, we first have to define the notion of an extraction function as well as some properties of extraction functions. For all of the following definitions let $\mathbb{T}$ denote the set of ontologies and let $\mathbb{M}$ denote the set of possible mappings between two ontologies $\mathcal{T}_1, \mathcal{T}_2 \in \mathbb{T}$.

Definition 1 (Extraction function). *A function $f : \mathbb{T} \times \mathbb{T} \times \mathbb{M} \rightarrow \mathbb{M}$ is an extraction function iff for all $\mathcal{T}_1, \mathcal{T}_2 \in \mathbb{T}$ and for all $\mathcal{M} \in \mathbb{M}$ we have $f(\mathcal{T}_1, \mathcal{T}_2, \mathcal{M}) \subseteq \mathcal{M}$.*

Some extraction functions solely rely on the input mapping $\mathcal{M}$, while some functions make use of the information encoded in $\mathcal{T}_1$ and $\mathcal{T}_2$. We take this distinction into account by refering to functions of the first type as ontology independent, while refering to functions of the second type as ontology dependent.

Definition 2 (Independence). *An extraction function f is ontology independent iff for all $\mathcal{T}_1, \mathcal{T}_2 \in \mathbb{T}$ and for all mappings $\mathcal{M} \in \mathbb{M}$ between $\mathcal{T}_1$ and $\mathcal{T}_2$ we have $f(\mathcal{T}_1, \mathcal{T}_2, \mathcal{M}) = f(\emptyset, \emptyset, \mathcal{M})$ where $\emptyset$ denotes an ontology with no axioms. Otherwise f is ontology dependent.*

In this work we focus on extraction functions that yield one-to-one mappings. A one-to-one mapping respectively a one-to-one extraction function can straight forward be defined as follows.

Definition 3 (One-to-one mapping and extraction function). *A mapping $\mathcal{M}$ is a one-to-one mapping iff for all correspondences $c \in \mathcal{M}$ there exists no $c' \neq c \in \mathcal{M}$ such that $source(c) = source(c') \vee target(c) = target(c')$. An extraction function f is a one-to-one extraction function iff the codomain of f is constrained to be the set of one-to-one mappings.*

As mentioned above, the extraction process can be understood as applying a sequence of extraction functions, in other words, applying the composition of several extraction functions. Since domain and codomain of an extraction function are not the same, we have to redefine composition with respect to extraction functions in the following self-evident way.

Definition 4 (Composition of extraction functions). *Given extraction functions f and g. The composition $f \circ g : \mathbb{T} \times \mathbb{T} \times \mathbb{M} \rightarrow \mathbb{M}$ is defined as $(f \circ g)(\mathcal{T}_1, \mathcal{T}_2, \mathcal{M}) = f(\mathcal{T}_1, \mathcal{T}_2, g(\mathcal{T}_1, \mathcal{T}_2, \mathcal{M}))$.*

Some matching systems are extracting the final mapping by applying a threshold on the similarity matrix. This way to extract is obviously not a one-to-one extraction. Nevertheless, it is an important component in the extraction process. Therefore, we define the application of a threshold as threshold extraction function.

Definition 5 (Threshold extraction function). *An extraction function f_t with $t \in [0, 1]$ is a threshold extraction function iff for all $\mathcal{T}_1, \mathcal{T}_2 \in \mathbb{T}$ and for all mappings $\mathcal{M} \in \mathbb{M}$ between $\mathcal{T}_1$ and $\mathcal{T}_2$ we have $f_t(\mathcal{T}_1, \mathcal{T}_2, \mathcal{M}) = \{c \in \mathcal{M} | conf(c) > t\}$.*

In most cases a one-to-one mapping will be obtained by the composition of a threshold function f_t and a one-to-one extraction function f. The order of applying these functions can have effects on the results depending on f. Therefore, we have to introduce the following property.

Definition 6 (Threshold commutative). *An extraction function f is threshold commutative iff $f \circ f_t = f_t \circ f$ for all $t \in [0, 1]$.*

Obviously, threshold extraction functions are threshold commutative. This follows from the fact that $f_{t_1} \circ f_{t_2} = f_{max\{t_1, t_2\}} = f_{t_2} \circ f_{t_1}$.

The most interesting class of extraction functions are, from a theoretical perspective, optimization functions. These functions are defined by an objective function and extract a final mapping that is optimal with respect to the chosen objective function.

Definition 7 (Optimization function). *An extraction function f_o is an optimization function iff for all $M^* \subseteq M$ we have $o(M^*) \leq o(f_o(\mathcal{T}_1, \mathcal{T}_2, M))$ with $o : \mathbb{M} \rightarrow \mathbb{R}$ being an objective function.*

We will see that the considerations of this section are not only of theoretical interest but constitute a useful framework to describe and distinguish between different extraction techniques.

3 One-to-one extraction algorithms

In this section we describe several algorithms that are implementations of different types of extraction functions. Some of these algorithms have to iterate over the elements of a mapping according to the ordering of confidence values. Therefore, we deal with mappings as sequences of correspondences in the following.

3.1 Ontology independent extraction algorithms

First, we focus on the algorithmic implementation of ontology independent extraction functions. These functions do not make any use of the information encoded in the ontologies to be aligned. At the end of this subsection we present a small example that compares the behavior of the introduced algorithms.

Algorithm 1

```
NaiveDescending($\mathcal{M}$)
$\mathcal{M}' \leftarrow \emptyset$
SortDescending($\mathcal{M}$)
**while** $\mathcal{M} \neq \emptyset$ **do**
    $c \leftarrow$ RemoveFirstElement($\mathcal{M}$)
    $\mathcal{M}' \leftarrow \mathcal{M}' \cup \{c\}$
    **for all** $c' \in$ GetAlternatives($\mathcal{M}, c$) **do**
        RemoveElement($\mathcal{M}, c'$)
    **end for**
**end while**
**return** $\mathcal{M}'$
```

Naive descending extraction Algorithm 1 can be described as naive greedy strategy that transforms a many-to-many mapping into a one-to-one mapping by iterating over the elements of $\mathcal{M}$ in descending order. The algorithm consists of a sequence of local decisions. First, the correspondences in $\mathcal{M}$ are sorted descending due to their confidence value. Then the algorithm iterates over $\mathcal{M}$ removing step by step elements from $\mathcal{M}$. In each iteration correspondence c with the hightest confidence value is removed and added to the extraction result $\mathcal{M}'$. The reduced $\mathcal{M}$ is checked for alternative correspondences, where GetAlternatives($\mathcal{M}, c$) is defined to return the set of all correspondences that have the same source or target concept in common with c. All alternatives are removed from $\mathcal{M}$ and will thus also not be contained in the extracted mapping $\mathcal{M}'$. Notice that this naive extraction function is threshold commutative and has been implemented by several participants of the OAEI 2006, for example by Falcon-AO [2] and RiMOM [4].

Naive ascending extraction Algorithm 2 is similar to algorithm 1 but more restrictive. While algorithm 1 accepts correspondences and removes their alternatives in descending order, algorithm 1 dismisses correspondences due to the existence of alternatives with a higher confidence value. The major difference is based on the principle that correspondences are dismissed if better alternatives exist even though these alternatives are also dismissed in one of the following iterations. Therefore, we have NaiveDescending($\mathcal{M}$) $\supseteq$ NaiveAscending($\mathcal{M}$) for all $\mathcal{M} \in \mathbb{M}$.

Hungarian extraction Algorithms 1 and 2 are implementations of extraction functions that are based on sequences of local decisions. Contrary to these approaches, an optimization extraction function extracts a solution that is optimal from a global point of view, selecting the one-to-one subset $\mathcal{M}'$ of $\mathcal{M}$ that is optimal with respect to some objective function o. Choosing objective function $o(\mathcal{M}') = \sum_{c \in \mathcal{M}'} conf(c)$, the problem to find $\mathcal{M}'$ can be solved using the hungarian method. The hungarian method is a combinatorial optimization algorithm which solves assignment problems in polynomial time [3]. To use the algorithm in the context of mapping extraction a few modifications have to be applied. Due to the lack of space we cannot give a detailed description. Such

Algorithm 2

```
NaiveAscending($\mathcal{M}$)
$\mathcal{M}' \leftarrow \emptyset$
SortAscending($\mathcal{M}$)
for all $c \in \mathcal{M}$ do
    if GetAlternatives($\mathcal{M}, c) = \emptyset$ then
        $\mathcal{M}' \leftarrow \mathcal{M}' \cup \{c\}$
    end if
end for
return $\mathcal{M}'$
```

a description can be found in [6] in section 3.1. We refer to this algorithm as HungarianExtraction(). Notice that the function implemented in HungarianExtraction() is our first example for a function that is not threshold commutative. A proof for this claim can be given by counter example. Example 1 provides a counter example and illustrates the differences between the three proposed extraction functions.

Example 1. Given a mapping $\mathcal{M} = \{c_1, c_2, c_3, c_4\}$ between $\mathcal{T}_1$ and $\mathcal{T}_2$ based on a similarity matrix S. The following table describes S respectively $\mathcal{M}$ in detail. Applying the extraction functions presented above we obtain different results.

	$2:X$	$2:Y$
$1:A$	$c_1 = \langle 1\colon A, 2\colon X, =, 0.9\rangle$	$c_2 = \langle 1\colon A, 2\colon Y, =, 0.8\rangle$
$1:B$	$c_3 = \langle 1\colon B, 2\colon X, =, 0.7\rangle$	$c_4 = \langle 1\colon B, 2\colon Y, =, 0.5\rangle$

- NaiveDescending($\mathcal{M}$) $= \{c_1, c_4\}$: The algorithms first adds c_1 to the resulting mapping $\mathcal{M}'$. By making this choice c_2 and c_3 are removed from $\mathcal{M}$ since they are alternatives with a lower confidence value. Finally, c_4 is added to $\mathcal{M}'$.
- NaiveAscending($\mathcal{M}$) $= \{c_1\}$: The algorithm starts with c_4. Since there are more probable alternatives available c_4 is discarded. The same holds for c_3 and c_2. Finally, only c_1 is left and accepted to be part of $\mathcal{M}'$.
- HungarianExtraction($\mathcal{M}$) $= \{c_2, c_3\}$: The hungarian method finds the best one-to-one mapping $\mathcal{M}'$ with respect to the objective function $o(\mathcal{M}) = \sum_{c \in \mathcal{M}'} c$. Notice that correspondence c_1 is not an element of $\mathcal{M}'$ contrary to the results of the greedy approaches.

What if we apply threshold function $f_{0.75}$ to this problem? Now we have HungarianExtraction($f_{0.75}(\mathcal{M})$) $= \{c_1\}$ and $f_{0.75}$(HungarianExtraction($\mathcal{M}$))$= \{c_2\}$. Thus, we can conclude that our implementation of an optimizing extraction function is not threshold commutative.

3.2 Ontology dependent extraction algorithms

We now introduce two straight forward extensions for each of the algorithms presented above. The resulting algorithms are implementations of ontology dependent extraction functions. First, we discuss how to use the additional information encoded in $\mathcal{T}_1$ and $\mathcal{T}_2$ by introducing the notion of a merged ontology (definition 8). Merging $\mathcal{T}_1$ and $\mathcal{T}_2$ results in the union of $\mathcal{T}_1$ and $\mathcal{T}_2$ using $\mathcal{M}$ as nexus.

Definition 8 (Merged ontology). *Given a mapping $\mathcal{M}$ between ontologies $\mathcal{T}_1$ and $\mathcal{T}_2$. The merged ontology $\mathcal{T}_1 \cup_{\mathcal{M}} \mathcal{T}_2$ of $\mathcal{T}_1$ and $\mathcal{T}_2$ connected via $\mathcal{M}$ is defined as $\mathcal{T}_1 \cup_{\mathcal{M}} \mathcal{T}_2 = \mathcal{T}_1 \cup \mathcal{T}_2 \cup \{t(c) \mid c \in \mathcal{M}\}$ with t being defined by $t(\langle 1\colon C, 2\colon D, =, c\rangle) = 1\colon C \equiv 2\colon D$ converting equivalence correspondences into equivalence axioms of $\mathcal{T}_1 \cup_{\mathcal{M}} \mathcal{T}_2$.*

Adding the correspondences of $\mathcal{M}$ as equivalence statements results in an ontology that is structured by subsumption relations connecting concepts of both ontologies. A concept of $\mathcal{T}_1$ is thus positioned in the taxonomy of $\mathcal{T}_2$ and vice versa. This can result in subsumptions between concepts of $\mathcal{T}_1$ or $\mathcal{T}_2$ that cannot be derived from $\mathcal{T}_1$ respectively $\mathcal{T}_2$ without taking $\mathcal{M}$ into account. We define the according property as instability of a mapping (also defined in [5] in the context of DDL).

Definition 9 (Stability of a mapping). *Given a mapping $\mathcal{M}$ between ontologies $\mathcal{T}_1$ and $\mathcal{T}_2$. $\mathcal{M}$ is stable iff there exists no pair of concepts $\langle i\colon C, i\colon D\rangle$ with $i \in \{1, 2\}$ such that $\mathcal{T}_i \not\models i\colon C \sqsubseteq i\colon D$ and $\mathcal{T}_1 \cup_{\mathcal{M}} \mathcal{T}_2 \models i\colon C \sqsubseteq i\colon D$. Otherwise $\mathcal{M}$ is instable.*

The additional subsumption statements introduced by a mapping are major topic of distributed description logics (compare for example [9]). In this context, additional subsumption statements are equated with additional knowledge which has a positive connotation. Nevertheless, we think that - as far as we are concerned with automatically generated correspondences - additional subsumption statements introduced by $\mathcal{M}$ indicate that some of the correspondences in $\mathcal{M}$ are erroneous. Furthermore, we introduce the stronger property of mapping consistency (based on the corresponding definitions in [10], [5], and [7]).

Definition 10 (Consistency of a mapping). *Given a mapping $\mathcal{M}$ between ontologies $\mathcal{T}_1$ and $\mathcal{T}_2$. $\mathcal{M}$ is consistent iff there exists no concept $i\colon C$ with $i \in \{1, 2\}$ such that $\mathcal{T}_i \not\models i\colon C \sqsubseteq \bot$ and $\mathcal{T}_1 \cup_{\mathcal{M}} \mathcal{T}_2 \models i\colon C \sqsubseteq \bot$. Otherwise $\mathcal{M}$ is inconsistent.*

Obviously, some of the correspondences of an inconsistent mapping $\mathcal{M}$ have to be incorrect, because we would not accept a mapping that imposes restrictions on $\mathcal{T}_1 \cup_{\mathcal{M}} \mathcal{T}_2$ making some of the concepts in $\mathcal{T}_1 \cup_{\mathcal{M}} \mathcal{T}_2$ unsatisfiable. By postulating that the extracted mapping $\mathcal{M}'$ has to be stable respectively consistent, we impose additional constraints on extraction functions. Notice that both properties are subject to criticism.

- Mapping stability: Extracting stable mappings makes only sense if all (or most of all) subsumption statements have been specified (directly or indirectly via entailment) in both $\mathcal{T}_1$ and $\mathcal{T}_2$. If one of the ontologies has been poorly structured by the ontology engineer the additional subsumption statements imposed by $\mathcal{M}$ will indeed introduce new acceptable knowledge.
- Mapping inconsistency: Inconstencies will only occur if $\mathcal{T}_1$ respectively $\mathcal{T}_2$ contain disjointness statements. Evaluation of ontologies show that these statements are often missing [13]. Thus, even a completely incorrect mapping will often be consistent.

Even though there are extraction problems where stability is too restrictive while consistency is not restrictive at all, we believe that applying these properties will in most cases have positive effects on the extraction results. Therefore, we introduce the notion of a minimal conflict set. A minimal conflict set is an instable respectively inconsistent subset of $\mathcal{M}$ which contains no real subset that is also instable respectively inconsistent.

Definition 11 (Minimal conflict sets). *Given a mapping $\mathcal{M}$ between ontologies $\mathcal{T}_1$ and $\mathcal{T}_2$. A subset $C \subseteq \mathcal{M}$ is a minimal conflict set with respect to stability (consistency), if C is instable (inconsistent) and each $C' \subset C$ is stable (consistent).*

From a more general point of view we can abstract from the conflicts introduced in this paper and define the class of conflict based extraction functions as subclass of ontology dependent extraction functions.

Definition 12 (Conflict based extraction function). *An extraction function f_g is conflict based iff for all $C \in g(\mathcal{M})$ there exists a correspondence $c \in C \wedge c \notin f_g(\mathcal{T}_1, \mathcal{T}_2, M)$ where $g : \mathbb{M} \rightarrow 2^{\mathbb{M}}$ is a conflict function defined by $g(\mathcal{M}) = \{C \subseteq \mathcal{M} | C$ is a minimal conflict set$\}$.*

Using a conflict based approach origins from its application in the context of diagnosis, as introduced by Reiter [8]. It has already been applied by the authors to the problem of automatically repairing mappings in [7] and partially to the problem of mapping extraction in [6]. Notice that related approaches can be found in [12] where Wang and Xu propose similar mapping properties. These properties are used supporting humans in semi-automatic repairing of mappings.

As decribed in [6] stability and consistency of $\mathcal{M}$ can be efficiently checked for each dual-element subset of $\mathcal{M}$ using a straight forward approach.[1] Therefore, we restrict ourselves to pairwise consistency and stability and have to define a conflict pair as a dual-element conflict set.

Definition 13 (Conflict pair). *Given a mapping $\mathcal{M}$ between ontologies $\mathcal{T}_1$ and $\mathcal{T}_2$. C is a stability (consistency) conflict pair, if $|C| = 2$ and C is a minimal conflict set with respect to stability (consistency). Correspondence $c_1 \in \mathcal{M}$ conflicts with correspondence $c_2 \in \mathcal{M}$ iff $\{c_1, c_2\}$ is a conflict pair.*

The extraction algorithms we propose as approximate conflict based extensions of algorithms 1 and 2 remove at least one correspondence from each conflict pair which results in an extracted mapping that will be pairwise stable respectively consistent. Both algorithms can be extended in a natural way by replacing GETALTERNATIVES($\mathcal{M}, c$) by calling a method that returns all alternatives for c as well as all correspondences that conflict with c. Obviously, the results of these extended algorithm are pairwise consistent respectively stable. For the experimental study we have implemented both variants. Notice that pairwise stability (consistency) is only an approximation of stability (consistency). Nevertheless, due to our experience most conflict sets turned out to be minimal conflict pairs.

The same approach cannot be applied to extend the hungarian method. Instead of that it is possible to search for an optimal solution that is free of pairwise conflicts. Due to the lack of space we have to omit a description of the algorithm which has already been described in detail in [6]. There we claimed that conflict based extraction not only increases precision but in some cases also increases recall. We have been critizised for this point of view, because it seems that solving a conflict is established

[1] In [6] this has only been described for consistency, but this approach can be modified in a self-evident way for stability.

by dismissing correspondences. That is only half the truth. Solving conflicts forces the hungarian method to rearranage parts of or even the whole assignment. Consider again example 1. Suppose now c_2 conflicts with c_3. Obviously, c_2 and c_3 cannot be both elements of the final extraction. But since alternatives are available, the final result of the extended hungarian extraction will be $\{c_1, c_4\}$ instead of $\{c_2, c_3\}$. Thus, it is possible that the conflict-based extraction increases both precision and recall. The same kind of argument can be applied to NAIVEDESCENDING() even if a rearrangement will only affect parts of the results due to the greediness of the algorithm. Contrary to this, increasing recall is not possible by extending algorithm 2. For this algorithm we have $\text{NAIVEASCENDING}(\mathcal{M}) \supseteq \text{NAIVEASCENDING}_{Consistency}(\mathcal{M}) \supseteq \text{NAIVEASCENDING}_{Stability}(\mathcal{M})$.

4 Experiments

In the following we present some emprirical results by applying the one-to-one extraction algorithms presented above on real world matching problems. Before going into details it makes sense to bring together the theoretical considerations of the last sections and resume them as hypotheses.

- H_1: One-to-one extraction functions exceed threshold extraction functions with respect to the harmonic mean of precision and recall (f-measure).
- H_2: Ontology dependent extraction functions (implemented as conflict-based algorithms) will result in better extractions compared to alternative approaches.
- H_3: Optimization extraction functions will result in better extractions compared to alternative approaches.

4.1 Experimental Settings

We evaluated the presented algorithms using automatically created mappings between ontologies of the ontoFarm data set. The ontoFarm data set consists of a set of ontologies in the domain of conference organization that has been created by the Knowledge Engineering Group at the University of Economics Prague [11] and has been subject of the conference track at the Ontology Alignment Evaluation Initiative 2006.

Amongst the participants that submitted results for all pairs of ontologies only two matching systems generated many-to-many mappings that can be used as input to our one-to-one extraction procedures. We refer to these systems as system A and B. We were also able to refactor another matching system (system C) to generate many-to-many mappings while not modifying any other parameters that have been used to generate the submitted results. Thus, we could apply and evaluate our algorithms as final step of three matching systems. To evaluate the extraction results we had to manually construct reference mappings consisting of equivalence correspondences for all pairs of ontologies. Since this task is extremly time-consuming we decided to choose a subset of eight ontologies of the ontoFarm Dataset, thus creating 28 reference mappings. In our experiments we evaluated the extraction algorithms presented above for all matching systems and for all matching problems. Thus we had to evaluate $9 \times 3 \times 28 = 756$ different extraction results. Therefore, we cannot present the results in detail but have to focus on aggregated values.

4.2 Results

To examine hypothesis H_1 we computed precision and recall aggregated over all matching problems for both the many-to-many input mappings and the extraction results based on applying the simplest one-to-one extraction algorithm NaiveDescending(). The resulting differences are listed in table 1. The extraction method NaiveDescending() is already implemented as extraction method of system C. Thus, we compare against the intermediary many-to-many mapping of this system. System A has been applied with a setting that results in an extraction close to a one-to-one mapping but also adds correspondences that have nearly (specified by a certain range) the same confidence as the best match. Therefore, for system A the effects of a one-to-one extraction are only marginal. System B does not extract one-to-one mappings. The resulting mapping can be optimized to a significant degree. While we gain about 26% precision, we loose only 4% recall. Nevertheless, even for system A we get slightly better results. We conclude that hypothesis H_1 has been verified by our experiments.

Matcher	Δ Precision	Δ Recall	Δ F-Measure
System A	+ 2.2%	0.0%	+ 1.3%
System B	+ 25.8%	- 3.9%	+ 13.3%
System C	+ 27.4%	- 3.0%	+ 16.6%

Table 1. Many-to many mappings vs. one-to-one mappings.

Hypothesis H_2 is concerned with the issue wether the information encoded in the ontologies can be used to optimize the extraction result. We proposed two ways to extend extraction functions to be ontology dependent, in particular conflict based. Thus, we compare for each extraction function the ontology independent algorithm with the variants generating consistent respectively stable extraction results. Figure 1 presents the mean values aggregating over matching systems and matching problems. Each column describes in how far precision, recall, and f-measure has been increased or decreased by applying one of the extraction methods compared to the many-to-many input mapping. For all three extraction methods and their extensions we can observe a similar pattern. The ontology independent variant is less precise than the ontology dependent variant using consistency which is again less precise than the variant using stability. A similar pattern with a negative influence of ontology dependency can be observed for recall. But notice that the negative effects are smaller compared to the positive effect on precision. This can also be derived from the third group of columns describing the f-measure. Only for algorithm NaiveDescending() we have slightly worse results by extending it to an ontology dependent approach. We have already argued above that extending this algorithm will not result in an rearrangement of the extracted assignment. We can conclude that hypotheses H_2 has been verified by our experiments.

Finally, we have to consider hypotheses H_3. We have claimed that an extraction function that finds an optimal extraction with respect to an appropriate objective function yields better results than a greedy approach. This hypothesis cannot be verified

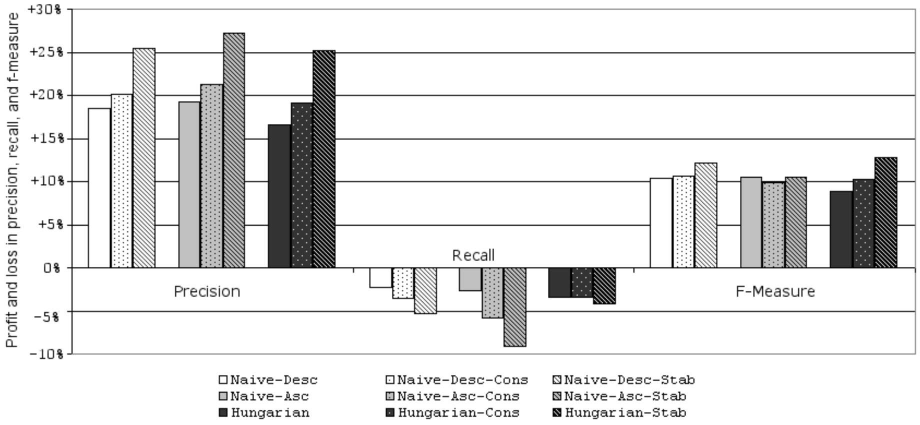

Fig. 1. Comparison of three basic extraction methods and their ontology dependent extensions.

by the experimental results. If we compare the f-measures of NaiveDescending() and HungarianExtraction() we observe that the greedy approach surprisingly extracts slightly better solutions in average. There is some evidence that choosing an additive objective function has not been the best choice. Understanding confidence values as probabilities, it seems to be more natural to use an objective function like $o(\mathcal{M}') = \prod_{c \in \mathcal{M}'} conf(c)$ which possibly will result in better extractions. But notice also that using additional conflict information has the strongest positive effect on the optimization approach. We have argued above that an optimization approach is more flexible with respect to rearrangements of assignments. Therefore, taking additional conflicts into account in the context of an optimization approach will strongly influcence the quality of the extraction results, while algorithm NaiveAscending() can use information on conflicts only to a very limited extent.

5 Summary and Outlook

We introduced a framework for describing and distinguishing between methods to extract a one-to-one mapping from a similarity matrix. In particular, we introduced the notion of an ontology dependent extraction function as well as the notion of an optimization extraction function. From a theoretical perspective we have argued that these types of extraction functions should result in better solutions to the extraction problem than naive approaches. We stated several algorithms as implementations of naive and more sophisticated extraction functions. To extend these algorithms towards ontology dependency, we introduced the concepts of mapping consistency and stability based on prior work. All in all, we stated nine different extraction methods. Thus, we had for each important type of extraction function at least one implementation.

In order to proove the hypothesis derived from theoretical considerations, we performed several experiments. In these experiments we could verify that ontology dependent functions yield better results than their ontology independent counterparts. We could not proove that our implementation of an optimization extraction function performs better than the naive approaches and argued that the choice of objective function is a crucial point. We also observed that extending optimization functions results in a significant improvement compared to the ontology independent counterparts. In future work the combination of optimization and conflict based extractions has to be examined. In particular, understanding confidences as probabilities seems to be an promising approach. Besides arguing that certain methods result in better solutions compared to naive approaches, the main contribution of this paper is to highlight the importance of the extraction problem as a self-contained subproblem of ontology matching.

Acknowledgement The work has been partially supported by the German Science Foundation (DFG) in the Emmy Noether Programme under contract STU 266/3-1.

References

1. Jerome Euzenat and Pavel Shvaiko. *Ontology Matching*. Springer Verlag, 2007.
2. Wei Hu, Gong Cheng, Dongdong Zheng, Xinyu Zhong, and Yuzhong Qu. The results of falcon-ao in the oaei 2006 campaign. In *Proceedings of the ISWC 2006 Workshop on Ontology Matching*, Athens, GA, USA, November 2006.
3. H.W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics*, pages 83–97, 1955.
4. Yi Li, Juanzi Li, Duo Zhang, and Jie Tang. Result of ontology alignment with rimom at oaei06. In *Proceedings of the ISWC 2006 Workshop on Ontology Matching*, Athens, GA, USA, November 2006.
5. Christian Meilicke. Reasoning about ontology mappings in distributed description logics. Bachelor Thesis, University of Mannheim, August 2006.
6. Christian Meilicke and Heiner Stuckenschmidt. Applying logical constraints to ontology matching. In *Proccedings of the 30th Annual German Conference on Artificial Intelligence (KI-07)*, Osnabrück, Germany, 2007.
7. Christian Meilicke, Heiner Stuckenschmidt, and Andrei Tamilin. Repairing ontology mappings. In *Proceedings of the Twenty-Second Conference on Artificial Intelligence (AAAI-07)*, Vancouver, Canada, 2007.
8. Ray Reiter. A theory of diagnosis from first principles. *Artificial Intelligence*, 32:57–95, 1987.
9. L. Serafini and A. Tamilin. DRAGO: Distributed reasoning architecture for the semantic web. In *Proceedings of the Second European Semantic Web Conference (ESWC'05)*, 2005.
10. Heiner Stuckenschmidt, Holger Wache, and Luciano Serafini. Reasoning about ontology mappings. In *Proceedings of the ECAI-06 Workshop on Contextual Representation and Reasoning*, August 2006.
11. Ondrej Svab, Svatek Vojtech, Petr Berka, Dusan Rak, and Petr Tomasek. Ontofarm: Towards an experimental collection of parallel ontologies. In *Poster Proceedings of the International Semantic Web Conference 2005*, 2005.
12. P. Wang and B. Xu. Debugging ontology mapping. In *International Workshop on Semantic e-Science (SeS2006) of the 1st Asian Semantic Web Conference (ASWC2006)*, 2006.
13. T. D. Wang. Gauging ontologies and schemas by numbers. In *Proceedings of the Workshop EON Evaluation of Ontologies for the Web*, 2006.