**Combination of Methods for Ontology Matching**

Tyl, Pavel
2008

# Combination of Methods for Ontology Matching

*Post-Graduate Student:*
ING. PAVEL TYL

Institute of Computer Science of the ASCR, v. v. i.
Pod Vodárenskou věží 2
182 07 Prague, Czech Republic

Faculty of Mechatronics and Interdisciplinary Engineering Studies
Technical University of Liberec
Hálkova 6
461 17 Liberec, Czech Republic

pavel.tyl@tul.cz

*Supervisor:*
ING. JÚLIUS ŠTULLER, CSC.

Institute of Computer Science of the ASCR, v. v. i.
Pod Vodárenskou věží 2
182 07 Prague, Czech Republic

stuller@cs.cas.cz

Field of Study:
Technical Cybernetics

This work was partly supported by the Research Center 1M0554 of Ministry of Education of the Czech Republic: "Advanced Remedial Technologies", project 1ET100300419 of the Program Information Society (of the Thematic Program II of the National Research Program of the Czech Republic): "Intelligent Models, Algorithms, Methods and Tools for the Semantic Web Realization" and by the Institutional Research Plan AV0Z10300504: "Computer Science for the Information Society: Models, Algorithms, Applications".

**Abstract**

While (partial) ontologies usually cover a specific topic/area, many applications require much more general approach to describe their data. Ontology matching can help to transform several such partial ontological descriptions into a single unifying one.

The paper describes a case study of using different methods, compares their advantages and discusses a possibility of using particular results for the definition of the final ontology. Two trivial ontologies were created (independently of any tool) and they were matched using various selected tools.

## 1. Introduction

Many ontologies were, and are, created in different areas of human activities. Ontologies often contain overlapping concepts. For example companies may want to use standard ontologies of certain domain community or authority along with ontology specific for their own company. In other words creators of ontologies can use existing ontologies as a basis for creating new ones by *integration* or *merging* of the *existing ones*.

*Ontology matching* is the process of finding relationships or correspondences between entities of different ontologies which are somehow semantically connected. The output of a matching process is a set of these correspondences between two or more ontologies called an *ontology alignment*. The oriented version of an ontology alignment is an *ontology mapping*[1]. Relationships originated by ontology matching can be used to realize the following operations on ontologies:

- *Ontology Merging*[2]– creating a new ontology containing concepts from source ontologies (in general overlapping – see Fig. 2). Initial ontologies (see Fig. 1) remain unaltered.

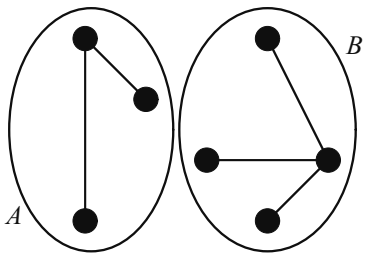


**Figure 1:** Initial ontologies *A* and *B*.

- *Ontology Integration* – inclusion of one ontology into another one by expressing the relationships between both of them, creating *"superontology"* connecting (partial) concepts and containing the knowledge from both source ontologies (see Fig. 3). One ontology remains unaltered while the other one is modified by knowledge of the first one.

[1] Ontology mapping can be seen as a collection of mapping rules (with some direction – from one ontology to another one, i.e. *Source* → *Target*).
[2] Ontology merging is similar to *schema integration* in databases.

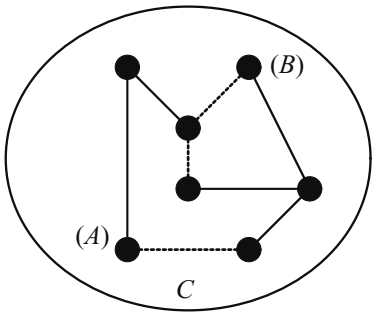


**Figure 2:** *Merging* – After merging the relationships between the original ontologies disappear.

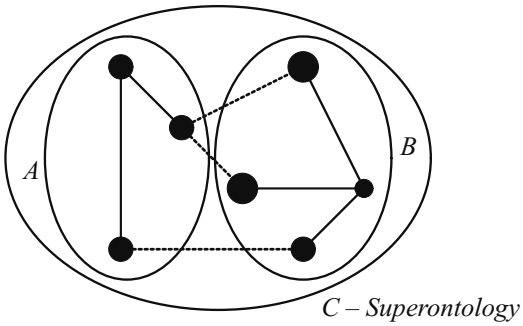


**Figure 3:** *Integration* – First ontology is unaltered while the second one is modified.

Whereas original ontologies are during *ontology merging* replaced by a new ontology (without initiation of direct correspondence between initial ones and the new one)[3], some documents need not reflect this replacement, but denote original ontologies. On the contrary, in the case of *ontology integration* the *superontology* is logicaly connected with the initial ontologies and in case some documents reference a concept from an original ontology, this concept is put over superontology. For this reason I prefer *ontology integration* in practise.

Ontology matching is in most cases done manually or semi-autoamtically, mostly with a support of some graphical user interface. A manual specification of ontology parts for matching is time consuming and moreover error prone process. Therefore there is a need for development of faster methods, which can process ontologies at least semi-automatically.

There are several tools that support user ontology matching. These tools use various techniques for proposal of integration rules, some advanced ones solve the question how to effectively combine results of particular techniques. These techniques unwind from the level of abstraction they work with.

Disadvantage of some of these methods is the necessity of setting numerous parameters from which sugesstions of integration rules unwind. In many of them the parameters setting plays so essential role that it can not be accomplished without deeper knowledge of concepts desribed in partial input ontologies.

Usually every matching tool innovates ontology matching on a particular aspect, nevertheless there exist several similar properties (with only minor exceptions) common to all of these tools [4]:

- Schema-based matching solutions are much more investigated than instance based solutions. This is partly caused by the fact instances may not be available during ontology matching process.
- Most of the systems focus on specific application domains (medicine, music...) as well as on dealing with particular ontology types (RDF, OWL...). Only few system are so general they can suit various application domains together with generic ones and support multiple formats. These are, for example, COMA++ [I2] or S-Match [5].
- Most approaches take as input a pair of ontologies. Only few systems take as input multiple ontologies or more general structures. These are, for example, DCM [6] or Wise-Integrator (automatical web form data integration) [7].
- Most of the approaches handle only tree-like structures. Several advanced systems handle more general graph structures. These are, for example, COMA/COMA++ [I2] or OLA [I3] (uses Alignment API [I1]).
- Most of the systems focus on discovering of *one-to-one* alignments. But it is possible to encounter more complicated relationships as *one-to-many* or *many-to-many*. These relationships can manage for example DCM (use statistical methods and is not applicable in this study) [6] or CTXMatch2 [1].
- Most of the systems identify relationships (i.e. Prompt [8]), some of them focus on computing confidence measures of these relationships (i.e. COMA++ [I2]). This is based on the assumption of equivalence relation between ontology entities. Only few systems compute logical relations between ontology entities (such as equivalence or subsumption). These are for example CTXMatch2 [1] or S-Match [5].

[3] Correspondences between ontologies, provenance and other metadata can be represented by other indirect methods [11].

## 2. Experiment

The folowing tools were used in the experiment as representatives of "exceptions" from the previous list – COMA++ [I2], CTXMatch [1] and Alignment API [I1]. For demonstration of automatical suggestion of alignment was used Prompt [8] (plugin for Protégé system [I6]).

Our experiments were executed with the test OWL [I4] ontologies (MyPerson.owl and MyCustomer.owl) pictured on Fig. 4. For testing the ontologies containing classes only were used.

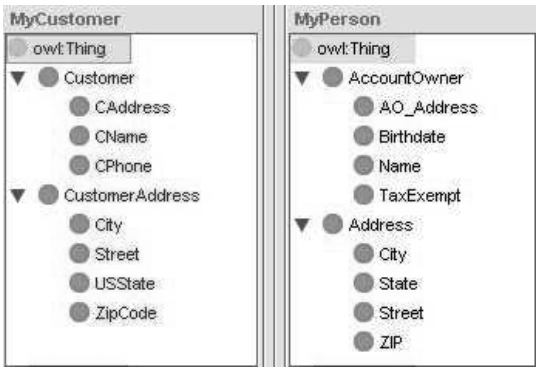


**Figure 4:** Test ontologies.

The test ontologies were matched directly by particular tools or by application interfaces.

Elements of the test ontologies were numbered in the following way:

| Columns | | Rows | |
|---|---|---|---|
| 1: | AccountOwner | 1: | CustomerAddress |
| 2: | AO_Address | 2: | Street |
| 3: | Birthdate | 3: | ZipCode |
| 4: | TaxExempt | 4: | City |
| 5: | Name | 5: | USState |
| 6: | Address | 6: | Customer |
| 7: | State | 7: | CPhone |
| 8: | Street | 8: | CName |
| 9: | City | 9: | CAddress |
| 10: | ZIP | | |

Following table represents relationships that could be subjectively expected as "ideal" on the assumption that *Account Owners* are considered to be *Customers* ($\sim$), etc. Sign $\sqsubset$ means the relation of subsumption. Sign $\sqsupset$ denotes generalization. Values in the tables then express a confidence measure of the fact that relations mentioned above conform. If there are some missing rows or columns in the tables, they contained no data.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | | | | | $\sim$ | | | | |
| 2 | | | | | | | | $\sim$ | | |
| 3 | | | | | | | | | | $\sim$ |
| 4 | | | | | | | | | $\sim$ | |
| 5 | | | | | | | $\sim$ | | | |
| 6 | $\sim$ | | | | | | | | | |
| 7 | | | | | | | | | | |
| 8 | | | | | $\sim$ | | | | | |
| 9 | | $\sim$ | | | | | | | | |

### 2.1. CTXMatch

CTXMatch2.2 [1] uses a semantic matching approach. It translates the ontology matching problem into the logical validity problem and computes logical relations, such as equivalence or subsumption between concepts and properties. CTXMatch is a sequential system which, at the element level, uses only WordNet [I7] to find initial matches for classes. At structure level it uses logical reasoners (i.e. Pellet [I5]) with the help of deductive techniques and verification of performability of logical formulas to compute resulting matching.

**Threshold value** – Matching results can be filtered by setting the **threshold** in the $< \mathbf{0}, \mathbf{1} >$ range. Relationships rated by lower value (in case of this experiment value 0.5) are not reflected (and not displayed in tables) for inconclusiveness.

**Ontology throughpass task** – Deep ontology throughpass (**hierarchical task**) is denoted by the word "hierarchy", flat throughpass (**flat task**) is denoted by the word "flat".

**Mapping** – Mapping **one-to-one** is denoted by **1:1**. Mapping **many-to-many** is denoted by **M:M**.

The same settings for all the experiments are the following:

- threshold: **0.5**
- input format: **OWL** [I4]
- output format: **XML** [I8]
- matching method: **DL** (using of description logic for deduction of possible relationships)

### 2.2. Alignment API

Alignment API is Java application interface that uses methods based on processing of word strings (String-based methods). It is used by other matching tools like OLA [I3] or FOAM [3].

**Levenshtein Algorithm** – The Levenshtein distance [9] is defined as the minimal number of characters we have to replace, insert or delete to transform one string into another.

**Smoa Algorithm** – Smoa [9] is the measure dependent on the length of "common" substrings and "not common" substrings, when the second mentioned part is deleted from the first one.

**WordNet Database** – WordNet [I7] is a leading linguistic database of English at worldwide scale. It groups together english words into the set of synonyms called *synsets* and give their short general definitions.

### 2.3. COMA++

COMA/COMA++ (COmbination of Matching Algorithms) [2] is a schema matching tool based on parallel composition of matchers. In his graphical user interface it offers an extensible libraries of matching algorithms. It is possible to modify default settings and parameters for certain ontologies in order to get better results. Parameters and settings in this case are not only threshold or one default string method, but many others (for example setting of consequence of used techniques).

### 2.4. Prompt

Prompt [8] is an extension plugin to the Protégé editor [I6]. Among other operations with pairs of ontologies (merging, extraction, versioning...) Prompt offers also interface for transformation of one ontology to another one and therefore it uses automatic matching at first.

**Table 1:** Similarity Measure CTXMatch – hierarchy – M:M.

| | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|
| 1 | | $\sqsupset$ **1.0** | $\sqsupset$ 1.0 | $\sqsupset$ 0.56 | $\sqsupset$ 0.56 | $\sqsupset$ 0.56 |
| 2 | | $\sqsupset$ 1.0 | $\sqsupset$ 1.0 | $\sqsupset$ **1.0** | $\sqsupset$ 0.56 | $\sqsupset$ 0.56 |
| 3 | | $\sqsupset$ 1.0 | $\sqsupset$ 1.0 | $\sqsupset$ 1.0 | $\sqsupset$ 0.56 | $\sqsupset$ **0.56** |
| 4 | | $\sqsupset$ 1.0 | $\sqsupset$ 1.0 | $\sqsupset$ 0.56 | $\sqsupset$ **1.0** | $\sqsupset$ 0.56 |
| 5 | | $\sqsupset$ 1.0 | $\sqsupset$ **1.0** | $\sqsupset$ 0.56 | $\sqsupset$ 0.56 | $\sqsupset$ 0.56 |
| 6 | | | $\sqsupset$ 0.67 | | | |
| 7 | | | $\sqsupset$ 0.67 | | | |
| 8 | $\sqsupset$ **0.67** | | $\sqsupset$ 0.67 | | | |
| 9 | | $\sqsupset$ 1.0 | $\sqsupset$ 1.0 | $\sqsupset$ 0.56 | $\sqsupset$ 0.56 | $\sqsupset$ 0.56 |

**Table 2:** Similarity Measure CTXMatch – hierarchy + Semantic Relation – M:M.

| | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|
| 1 | | $\sqsupset$ **1.0** | $\sqsupset$ 1.0 | $\sim$ 0.48 | $\sim$ 0.48 | $\sim$ 0.48 |
| 2 | | $\sqsupset$ 1.0 | $\sqsupset$ 1.0 | $\sqsupset$ **1.0** | $\sim$ 0.4 | $\sim$ 0.4 |
| 3 | | $\sim$ 0.62 | $\sim$ 0.62 | $\sim$ 0.4 | $\sim$ 0.4 | $\sim$ **0.4** |
| 4 | | $\sqsupset$ 1.0 | $\sqsupset$ 1.0 | $\sim$ 0.4 | $\sqsupset$ **1.0** | $\sim$ 0.4 |
| 5 | | $\sqsupset$ 1.0 | $\sqsupset$ **1.0** | $\sim$ 0.4 | $\sim$ 0.4 | $\sim$ 0.4 |
| 6 | | | $\sim$ 0.53 | | | |
| 7 | | | $\sim$ 0.45 | | | |
| 8 | $\sim$ **0.45** | | $\sim$ 0.45 | | | |
| 9 | | $\sqsupset$ 1.0 | $\sim$ 0.62 | $\sim$ 0.4 | $\sim$ 0.4 | $\sim$ 0.4 |

**Table 3:** Similarity Measure CTXMatch – hierarchy – 1:1.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| 1 | | | | | | $\sqsupset$ **1.0** | | | |
| 2 | $\sqsupset$ 0.39 | | | | | | | | |
| 3 | | | | | | | $\sqsupset$ 1.0 | | |
| 4 | | | | | | | | $\sqsupset$ 0.56 | |
| 5 | | | | | | | | | $\sqsupset$ 0.56 |
| 6 | | | | $\sqsubset$ 0.39 | | | | | |
| 7 | | | | | | | | | |
| 8 | | | | | $\sqsupset$ **0.67** | | | | |
| 9 | | | | | | | | | |

**Table 4:** Similarity Measure CTXMatch – hierarchy + Semantic Relation – 1:1.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| 1 | | | | | | $\sqsupset$ **1.0** | | | |
| 2 | $\sim$ 0.31 | | | | | | | | |
| 3 | | | | | | | $\sim$ 0.62 | | |
| 4 | | | | | | | | $\sqsupset$ 0.4 | |
| 5 | | | | | | | | | $\sqsupset$ 0.4 |
| 6 | | | | $\sqsubset$ 0.31 | | | | | |
| 7 | | | | | | | | | |
| 8 | | | | | $\sqsupset$ **0.45** | | | | |

**Table 5:** Similarity Measure CTXMatch – flat – 1:1.

| | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|
| 1 | | | | | |
| 2 | | | | $\sqsubset \sqsupset \sim$ **1.0** | |
| 3 | | | | | |
| 4 | | | | | $\sqsubset \sqsupset \sim$ **1.0** |
| 5 | | | | | |
| 6 | | | | | |
| 7 | | | $\sqsupset$ 1.0 | | |
| 8 | $\sqsupset$ **1.0** | | | | |

**Table 6:** Similarity Measure CTXMatch – flat + Semantic Relation – 1:1.

| | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|
| 1 | | | | | |
| 2 | | | | *Equiv.* **1.0** | |
| 3 | | | | | |
| 4 | | | | | *Equiv.* **1.0** |
| 5 | | | | | |
| 6 | | | | | |
| 7 | | | $\sqsupset$ 1.0 | | |
| 8 | $\sqsupset$ **1.0** | | | | |

**Table 7:** Similarity Measure CTXMatch – flat – M:M.

| | 5 | | 6 | | 7 | | 8 | 9 |
|---|---|---|---|---|---|---|---|---|
| 1 | | | $\sqsupset$ **1.0** | $\sim$ **0.7** | $\sqsupset$ 1.0 | $\sim$ 0.7 | | |
| 2 | | | | | $\sqsupset$ 1.0 | $\sim$ 0.7 | $\sqsubset \sqsupset \sim$ **1.0** | |
| 3 | | | | | $\sqsupset$ 1.0 | $\sim$ 0.7 | | |
| 4 | | | | | $\sqsupset$ 1.0 | $\sim$ 0.7 | | $\sqsubset \sqsupset \sim$ **1.0** |
| 5 | | | | | $\sqsupset$ **1.0** | $\sim$ **0.7** | | |
| 6 | | | | | $\sqsupset$ 1.0 | $\sim$ 0.7 | | |
| 7 | | | | | $\sqsupset$ 1.0 | $\sim$ 0.7 | | |
| 8 | $\sqsupset$ **1.0** | $\sim$ **0.7** | | | $\sqsupset$ 1.0 | $\sim$ 0.7 | | |
| 9 | | | $\sqsupset$ 1.0 | $\sim$ 0.7 | $\sqsupset$ 1.0 | $\sim$ 0.7 | | |

**Table 8:** Similarity Measure Alignment API – Levenshtein.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | $\sim$ **0.53** | | | | | | | | |
| 2 | | | | | | | | $\sim$ **1.0** | | |
| 3 | | | | | | | | | | $\sim$ **0.43** |
| 4 | | | | | | | | | $\sim$ **1.0** | |
| 5 | | | | | | | $\sim$ **0.71** | | | |
| 6 | | | | $\sim$ 0.5 | | | | | | |
| 7 | $\sim$ 0.33 | | | | | | | | | |
| 8 | | | | | $\sim$ **0.8** | | | | | |
| 9 | | | | | | $\sim$ 0.87 | | | | |

**Table 9:** Similarity Measure Alignment API – Smoa.

| | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|
| 1 | | $\sim \mathbf{0.82}$ | | | | |
| 2 | | | | $\sim \mathbf{1.0}$ | | |
| 3 | | | | | | $\sim \mathbf{0.6}$ |
| 4 | | | | | $\sim \mathbf{1.0}$ | |
| 5 | | | $\sim \mathbf{0.92}$ | | | |
| 6 | | | | | | |
| 7 | | | | | | |
| 8 | $\sim \mathbf{0.89}$ | | | | | |
| 9 | | $\sim 0.93$ | | | | |

**Table 10:** Similarity Measure Alignment API – WordNet.

| | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|
| 1 | | $\sim \mathbf{0.64}$ | | | | |
| 2 | | | | $\sim \mathbf{1.0}$ | | |
| 3 | | | | | | $\sim \mathbf{0.86}$ |
| 4 | | | | | $\sim \mathbf{1.0}$ | |
| 5 | | | $\sim \mathbf{0.83}$ | | | |
| 6 | $\sim 0.33$ | | | | | |
| 7 | | | | | | $\sim 0.6$ |
| 8 | $\sim \mathbf{0.94}$ | | | | | |
| 9 | | $\sim 0.97$ | | | | |

**Table 11:** Similarity Measure COMA++ – Own settings + COMA defaults.

| | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|
| 1 | | $\sim \mathbf{0.69}$ | | | | |
| 2 | | $\sim 0.77$ | | $\sim \mathbf{0.78}$ | | |
| 3 | | | | | | |
| 4 | | | | | | |
| 5 | | | $\sim \mathbf{0.83}$ | | | $\sim 0.61$ |

**Table 12:** Similarity Measure Prompt – Automatic Matching.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | | | | | | | | | |
| 2 | | | | | | | | $\sim$ | | |
| 3 | | | | | | | | | | |
| 4 | | | | | | | | | $\sim$ | |
| 5 | | | | | | | $\sim$ | | | |
| 6 | | | | | | | | | | |
| 7 | | | | | | | | | | |
| 8 | | | | | $\sim$ | | | | | |
| 9 | | $\sim$ | | | | $\sim$ | | | | |

## 3. Experiment evaluation

Mapping returned by tool CTXMatch with hierarchical throughpass task identified 6 from 8 possible relationships, but how it is visible from Table 1, next to these relationships are with the same coefficients of confidence detected other relationships between ontologies, which do not correspond to any facts. In other words it could be better to use this method for weighing of already detected relationships than for detection alone.

If we combine similarity measure with linguistic analysis (Semantic Relation), see Table 2, by most of wrong selected candidates comes to a downtrend of rating. This downtrend is noticed even by two nonconflicting rules, but without a detriment to correctness.

In case of choosing one-to-one mapping by the same method, we can do the selection of candidates in Table 3, respectively with linguistic analysis in Table 4 based on ratings from Table 1 and 2. By this selection initially

wrong selected relationships are reduced , but only 2 selected relationships correspond to the facts. While flat throughpass task (see Tab. 5) detected correctly 3 of 8 relationships, only one wrong relationship was selected by on-to-one mapping. The lexical analysis (see Tab. 6) does not bring any changes into the result mapping.

If we pass over the necessity of selection one-to-one mapping (as shown in Tab. 7), ambiguity of assigning ontology elements appears only by element 7 – *State*.

If we compare hierarchical throughpass task and flat throughpass task, selection of candidates is more restrictive (candidates have lower rating). If we focus on the type of analysis, then methods based on Levenshtein algorithm were able to correctly select 6 of 8 relationships with 3 wrong (see Tab. 8), methods based on Smoa algorithm selected correctly 6 relationships and 1 wrong (see Tab. 9). Methods using WordNet database selected 6 correct and 3 wrong relationships (see Tab. 10). If we compare results from these analysis with Table 1, we can see that relationships selected by these methods could help to solve ambiguities of selection (i.e. by tool CTXMatch). Not least COMA++ tool detected correctly 5 of 8 relationships (see Tab. 11) and it can be rated very positively without selection of wrong relationship. Similarly, extension Prompt, detected only 3 equivalences (see Tab. 12) without marking wrong relationship. Both last mentioned tools use combination of different methods and in term of confidence return correct mapping rules, but at the price of detecting incomplete set of these rule (some relationships are not detected at all).

## 4. Conclusion

In our experiment evaluation we have not found any tool that perfectly covers whole spectrum of ontology matching tasks. It tends to necessity of using more tools and combine their results.

From the performed study it follows (using given tools) that it may be appropriate to make rough outline by tool CTXMatch that can filter out candidates with less support. Ambiguity of selection can be solved by String-based methods (WordNet, Levenshtein Algorithm...), which do not need to give correct results on principle. Remarkable is really sporadic occurence of subsumption. It can be explained by the fact that subsumption can be detect in most cases by logical reasoner, when it uses information about existence of relationships between some concepts.

Our experiments reflect that tools COMA++ and Prompt offer better portfolio of methods, which are combined and return more accurate (but sometimes conservative) results with the possibility that some acceptable mappings are not proposed at all.

Therefore it is very useful to validate the results of matching process against the data used by initial ontologies. My future research direction will follow the same topics.

## References

[1] BOUQUET, P. – SERAFINI, L. – ZANOBINI, S.: “Semantic Coordination: A New Approach and Application”. In *Proc. 2$^{nd}$ International Semantic Web Conference (ISWC)*, volume 2870 of Lecture Notes in Computer Science, p. 130–145, Sanibel Island (FL US), 2003.

[2] DO, H. – RAHM, E.: “COMA – A System for Flexible Combination of Schema Matching Approaches”. In *Proc. 28$^{th}$ International Conference on Very Large Data Bases (VLDB)*, p. 610–621, Hong Kong (CN), 2002.

[3] EHRIG, M. – SURE, Y.: “FOAM – Framework for Ontology Alignment and Mapping – Results of the Ontology Alignment Evaluation Initiative”. In *Proceedings K-CAP Workshop on Integrating Ontologies*, volume 156, p. 72–76, Banff (CA), 2005.

[4] EUZENAT, Jérôme – SHVAIKO, Pavel: “Ontology Matching”. Springer-Verlag, Berlin/Heidelberg, 2007. ISBN 978-3-540-49611-3.

[5] GIUNCHIGLIA, F. – SHVAIKO, P. – YATSKEVICH, M.: “S-Match: An Algorithm and an Implementation of Semantic Matching”. In *Proc. Dagstuhl Seminar*, Internationales Begegnungs- und Forschungszentrum fuer Informatik (IBFI), Schloss Dagstuhl (DE), 2005.

[6] HE, B. – CHANG, K. C.: “Automatic Complex Schema Matching Across Web Query Interfaces: A Correlation Mining Approach”. *Volume 31 of ACM Transactions on Database Systems (TODS)*, p. 346–395, ACM, New York, 2006.

[7] HE, H. – MENG, W. – YU, C. – WU, Z.: “WISE-Integrator: A System for Extracting and Integrating Complex Web Search Interfaces of the Deep Web”. In *Proc. 31$^{st}$ International Conference on Very Large Data Bases (VLDB)*, p. 1314–1317, Trondheim (NO), 2005.

[8] NOY, F. N. – MUSEN, M.: “PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment”. In *Proc. 17$^{th}$ National Conference of*

*Artificial Intelligence (AAAI)*, p. 450–455, Austin (TX US), 2000.

[9] STOILOS, G. – STAMOU, G. – KOLLIAS, S.: "A String Metric for Ontology Alignment". In *Proc. 4$^{th}$ International Semantic Web Conference (ISWC)*, volume 3729 of Lecture Notes in Computer Science, p. 624–637, Galway (IE), 2005.

[10] STRACCIA, U. – TRONCY, R.: "oMAP: Combining Classifiers for Aligning Automatically OWL Ontologies". *Volume 3806 of Lecture Notes in Computer Science*, p. 133–147, Springer-Verlag, Berlin/Heidelberg, 2005.

[11] VRANDECIC, D. – VÖLKER, J. – HAASE, P. – TRAN, D. T. – CIMIANO, P.: "A Metamodel for Annotations of Ontology Elements in OWL DL". In *Proceedings of the 2$^{nd}$ Workshop on Ontologies and Meta-Modeling*, Karlsruhe (GE), 2006.

[I1] Alignment API and Alignment Server [online]: <http://alignapi.gforge.inria.fr>.

[I2] COMA++ – A System for Flexible Combination of Matching Algorithms [online]: <http://dbs.uni-leipzig.de/en/Research/coma.html>.

[I3] OLA – OWL Lite Alignment [online]: <http://www.iro.umontreal.ca/~owlola/alignment.html>.

[I4] OWL – Web Ontology Language / W3C Semantic Web Activity [online]: <http://www.w3.org/2004/OWL>.

[I5] Pellet – OWL Reasoner [online]: <http://pellet.owldl.com>.

[I6] Protégé – Ontology Editor and Knowledge Acquisition System [online]: <http://protege.stanford.edu>.

[I7] WordNet – Lexical database [online]: <http://wordnet.princeton.edu>.

[I8] XML – Extensible Markup Language / W3C XML Activity [online]: <http://www.w3.org/XML>.