# Semantic-based Identity Resolution and Merging for Business Intelligence

Milena Yankova and Horacio Saggion and Hamish Cunningham

Department of Computer Science, University of Sheffield
Regent Court, 211 Portobello Street,
Sheffield, S1 4DP
United Kingdom
{milena,saggion,hamish@dcs.shef.ac.uk

**Abstract.** In the context of ontology-based information extraction (OBIE), identity resolution is the process of deciding whether an instance extracted from text refers to a known entity in the target domain (e.g. the ontology). We have developed a customizable rule-based framework for identity resolution and merging which uses an ontology for knowledge representation. We present experiments on semantic-based identity resolution in the context of an OBIE system. The system for information extraction is a rule-based system which identifies conceptual information expressed in the domain ontology and it is based on a generic and adaptable human language technology. In the experiments we extract company information from several sources and update the ontology with the solved entities. Positive evaluation results show the interest of the undertaken approach.

## 1 Introduction

The MUSING project is integrating Human Language Technology and Semantics in the context of Business Intelligence (BI) applications. BI is the process of finding, gathering, aggregating, and analyzing information for decision making processes. Most BI systems are portals of information which facilitate business analysts the tasks of document search and navigation, however it is up to the user of the system to dig into huge amounts of unstructured information to find relevant facts to feed decision making processes such as company credit rating, measuring probability of success in particular business, finding an appropriate business partners, or getting up-to-date facts about business entities such as companies, places, and people. One solution to the manual extraction of facts from unstructured sources is to apply text processing and Natural Language Processing (NLP) techniques to automatically transform unstructured sources into structured representations suitable for such decision making processes.

Information Extraction (IE) is a key NLP technology which automatically extracts specific types of information from text to create records in a database or populate knowledge bases.

In MUSING, we work in the context of Ontology-based Information Extraction (OBIE) which is the process of identifying in text relevant concepts, properties, and relations expressed in an ontology of a particular application domain. In the context of ontology-based information extraction, one fundamental problems to be addressed is that of identification and merging of instances extracted from multiple sources. This process aims at identifying newly extracted (e.g. from text) facts and linking them to their previous mentions. Unlike classical information extraction (see [**?**]) where the extracted facts are only classified as belonging to pre-defined types, in an OBIE system, identity resolution aims at establishing a *reference link* between an object residing in the system's knowledge base and its mention in context (e.g. text).

This paper presents experiments on identity resolution using a general and adaptable framework. Recognizing identical or similar information across different sources is of paramount importance and in particular can lead to improved extraction performance from single sources.

Aggregation of extracted information has many advantages such as: complementing partial information from one source, increase extraction confidence, and keep updated information in knowledge bases.

Here we will introduce our Identify Resolution Framework (IdRF) which provides infrastructure for resolving identity of different classes of entities (e.g. organization, location, people). The framework uses the target ontology as an internal knowledge representation that provides detailed entity description formalism complemented with semantics. The framework is adaptable to different application domains and tasks.

## 2 Information Extraction

Information extraction is the process of extracting from text specific facts in a given target domain [Grishman 1996]. For example, in extracting information about companies key elements to be extracted are the company address, contact phone, fax numbers, and e-mail address, products and services, members of the board of directors and so on. The information to be extracted is pre-specified and the system is tailored to extract those specific elements. The field of information extraction has been fuelled by two major US international evaluations efforts, from 1987 until 1997 the Message Understanding Conferences [Grishman and Sundheim 1996] and since 2000 the Automatic Content Extraction Evaluation [ACE 2000].

Tasks carried out during information extraction are named entity recognition, which consists on the identification and classification of different types of names in text; coreference resolution, which is the task of deciding if two linguistic expressions refer to the same entity in the discourse; semantic role recognition, which deals with the recognition of semantic roles to sentence constituents (e.g. agent, goal); and relation extraction, which identifies relations between entities in text. Information extraction usually employs the following natural language processing technologies: parts-of-speech taggers, morphological analyser, named

entity recognisers, full (or partial) parsing, and semantic interpretation including nominal and verb coreference. These linguistic processors are generally available although some may require domain adaptation, for example while a parts-of-speech tagger for English could be used without major need for adaptation, a named entity recogniser should usually need adaptation to a new application domain.

There are two main approaches to the development of IE systems: (i) Handcrafted systems which rely on language engineers to design lexicons and rules for extraction, and (ii) machine learning systems which can be trained to perform one or more of the IE tasks. Learning systems are given either an annotated corpus for training or a corpus of relevant and irrelevant documents together with only a few annotated examples of the extraction task, in this case some non-supervised techniques such as clustering can also be applied.

Rule-based systems can be based on gazetteer lists - lists of keywords which can be used to identify known names (e.g. New York) or give contextual information for recognition of complex names (e.g. Corporation is a common postfix for a company name).- and cascades of finite state transducers which implement pattern matching algorithms over linguistic annotations (produced by various linguistic processors) [Cunningham et al 2002, Appelt et al, 1993]

Symbolic learning techniques which learn rules or dictionaries for extraction have been applied in information extraction. The AustoSlog system [Riloff, 1993] (and later the AutoSlog-TS system) automatically constructs a dictionary of extraction patterns using instantiation mechanism based on a number of syntactic templates manually specified, a corpus syntactically parsed, and a set of target noun phrases to extract. LP2 [Ciravegna, 2001] identifies start and end semantic tags using a supervised approach. LP2 learns three types of rules: tagging rules, contextual rules, and correction rules. The key of the process is in the separation of the annotations in start and end annotations and in the exploitation of the interactions between rules which identify start and end annotations. ExDISCO [Yangarber et al, 2000] learns extraction patterns (which then have to be associated with templates slots) from a set of documents. Statistical machine learning approaches to information extraction include the use of Hidden Markov Models (HMM); Support Vector Machines (SVM), and Conditional random Fields (CRF). With HMM the information extraction task is cast as a tagging problem where given a sequence of input words he system has to produce a sequence of tags and where the words are observations and the tags are hidden states in the HMM. SVM are very competitive supervised models for information extraction [Isozaki and Kazawa, 2002]. SVM cast the IE task as a binary classification problem (each label give rise to a binary classification problem). SVMs try to find an hyperplane in the vector space of instances that maximally separates positive from negative instances. Finding the hyperplane corresponds to an optimisation problem. CRFs [Lafferty et al, 2001] are state of the art techniques for IE and tend to do better than other classification methods.

In our work we are mainly interested in Ontology-based information extraction wich aims at identifying in text concepts and instances from an underlying

domain model specified in an ontology. In the following subsection we describe our approach.

### 2.1 MUSING Information Extraction

Our ontology-based information extraction system has been developed with the GATE platform which provides a set of tools for development of information extraction applications. In particular GATE provides support to work with ontologies. One of MUSING prototypes is a system for the extraction of company information, yet another prototype extracts region information. This information has to be extracted from many different sources such as web pages, financial news, and structured data sources. After extraction, the information is used for ontology population. Concepts targeted by the application are the company name, its main activities, its number of employees, its board of directors, etc. The extraction prototype uses some default linguistic processors from GATE, however the core of the system, the concept identification program was developed specifically for this application. In addition to specific processes of phrase chunking, lexicons and gazetteer lists have been created to perform gazetteer lookup processes. Rules for concept identification have been specified in regular grammars implemented in the JAPE language. A key element in the annotations created by the system is the encoding of ontological information - our applications create *Mention* annotations which make reference to the target ontology as well as the ontological concept a string of text refers to.

Figure 1 shows the automatic annotation of concepts in text. It is shown pieces of text annotated with ontological information. Note that the figure shows a semi-structured web page, also note that non-structured information is also targeted. As an example Figure shows rules for the identification of products and services in non-structured sources.

The result of the automatic annotation is further analysed by (i) a module which produces RDF triples associating different pieces of information together (e.g. a company with its number of employees, a company with its CEO), and (ii) the ontology population module responsible for knowledge base population. An evaluation of the performance of the extraction system indicates good results with over 84% F-score [**?**,**?**].

## 3 Identity Resolution Framework

IdRF provides a general solution to the identitication of known and new facts in particular domains. It can be used in different applications regardless of their particular domain or type of entity which need to be resolved. The input to IdRF is an entity together with its associated properties and values, the output is an integrated representation of the entity which will have new properties and values in the ontology.

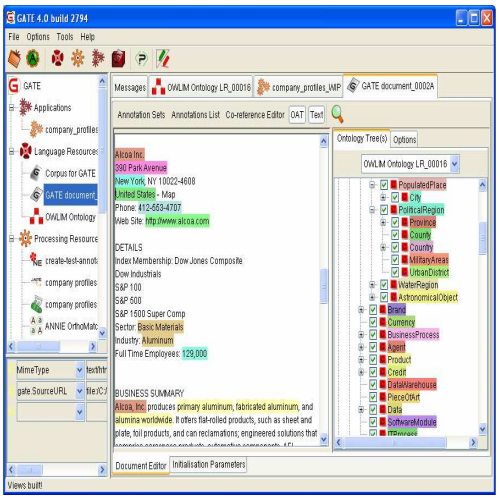


**Fig. 1.** Screenshoot of text processed with the OBIE system

A customisable identity criteria is in place to decide on the similarity between two instances. This criteria uses ontological operations and similarity computation between extracted and stored values which are weighted. The weighting criteria is specified according to the type of entity and the application domain.

**Fig. 2.** IdRF Architecture

### 3.1 Knowledge representation

The IdRF uses an ontology for internal and resulting knowledge representational formalism. The ontology not only contains the representation of the domain, but also known entities and properties. After identity resolution, the ontology Knowledge Base (KB) will contain entities with their full semantic description aggregated during the resolution process.

As a side effect of this continuous updating of the KB, the identity criteria is refined, thus improving the identity resolution by both refining the evidence calculation and introducing new entities serving as identity goals. Details about the two effects are given below:

– The evidence calculation is refined when a new value, attribute, property or relation is added to an existing instance description. Then, the identity

criteria for this instance is changed in order to reflect the newly available data adding new comparison restrictions. For example if the person age is added to his/her description, the age restriction will be added a new identity criterion.
- New entities added to the knowledge base represent potentially new goals for resolution. They are created by insertion of entirely new objects to the KB. When entities are processed in a later stage, they have to be compared not only two the previously available entities but also to the newly added instances.

Current implementation of the IdRF is based on the PROTON [**?**] ontology, which can be easily extended for any particular domain or specific task as it has been for the MUSING project. The knowledge base that actually contains the ontology and the instances associated with it is stored in the semantic repository provided by KIM [**?**] that is based on OWLIM [**?**] and Sesame.

### 3.2 IdRF Main Components

The IdRF framework receives an instance (e.g. type of instance and properties and values) and updates the ontology either asserting a new instance with its properties or updating an already existing instance. The IdRF architecture (see Figure 2) consists of four main stages.

- **Pre-filtering** - It filters out the irrelevant part of the ontology and forms a smaller set of instances similar to the source entity. It is intended to restrict the whole amount of ontology instances to a reasonable small number, to which the source entity will be compared. It can be regarded as pre-selection of ontology objects that are eligible to identification. The selected instances are potential target instances that might be identical to the source object; they already appear in the knowledge base and are somehow similar to the source object. Pre-filtering is realised by the Semantic Description Compatibility Engine (SDCE) which is described in details later on.
- **Evidence Collection** - It collects as much as possible evidence about the similarity between the source entity and each of the targets in the ontology. A set of similarity criteria is computed by SDCE by comparing corresponding attributes in the entity descriptions. Different comparison criteria are possible: some are based on string representation e.g. text edit distance, inverted frequency based matching; others can be web appearance, context similarity, etc.
- **Decision Maker** - Once all the evidences for different identity possibilities are collected, it concludes which is the best identity match. It is this third stage that decides about the strength of the presented evidence and makes the decision. This module chooses the candidate favoured by the class model natively stored as part of the Class Model described in SDCE. The models are based on the weighting of evidences. The model can be easily tuned by domain experts.

– **Data Integration** - After the decision is made the incoming entity is registered to the ontology as a final stage in the IdRF. The source entity can be either new one or successfully identified with an existing instance. If the system is not able to find a reliable match, the incoming object is inserted as a new instance in the KB. In case it is associated with an existing instance, then the object description is added to the description of the identified KB instance. Thus, the result from the current identification is stored in the ontology and is used for further identity resolution of the next incoming objects.

### 3.3 Semantic Description Compatibility Engine

Semantic Description Compatibility Engine (SDCE) is the evaluation engine of the IDRF. It is used in two stages of the framework pipeline: in the Prefiltering process it provides access to the ontology retrieving potential matching candidates according to a predefined class model rules; in the Evidence Collection stage it evaluates the similarity of potential matching pairs using corresponding class criteria.

**Class Models declaration** Execution of the SDCE is based on so-called class models that handle the specificity of different entity types presented as ontology classes. Given that the instances of different ontology classes diverge in their meaning and type, the SDCE class models formally describe them. These models are configured as formulas that express different conditions for candidates retrieval and comparison during the identification process.

The SCDE parser associates a formula with a specific classes in the ontology. The engine is based on first order probabilistic logic calculus and each class model is expressed by a single formula. Thus, each formula encodes the specificity of the corresponding class forming its model. Rule inheritance between classes is also supported allowing the set of formulas to be easily expanded for a new class. This is especially useful when the ontology is extended and refined or the focus of the particular IdRF application has been changed.

The formulas so far are valid only when the two instances are of one and the same class and in this case for each class we can define a single equivalence formula. When the two instances are from different classes then we have to use the formula that is attached to the most specific class common for both of them. In case if one of the classes is subclass of the other e.g.

$$\text{class(C1)\&class(C2)\&subClassOf(C2,C1)} \tag{1}$$

and we have equivalence formulas for both classes C1 and C2 then we should consider both of them i.e. the total confidence should be:

$$\text{formula for C1} \vee \text{formula for C2} \tag{2}$$

**Formulas** The formulas are described by predicates from a common pool of predicates where each primitive predicate is implemented as Java class, so the set of predicates is extensible using Java programming language. It is essential that several formulas can use one and the same predicate as part of their definitions. Since each primitive predicate is implemented as Java class. The idea of defining a number of simple predicates instead of a single complex one follows the library like or "code reuse" approach in software development. This allows us to support an extendable set of reusable primitive predicates from which someone can compose complex formulas in a declarative way.

Formulas are composed by combining predicates by the usual logical connectives like like "&", "|", "not" and "⇒". Common usage of the predicates is assisted also by the fact that a predicate can be weighted differently in different formulas according to its importance for the particular class model. This flexibility is achieved by different weights - real values from 0 to 1 - that can be attached to each of the predicates in the formulas using the logical connective "&".

*Example of formula definition*

```
namespace: rdf: "http://www.w3.org/1999/02/22-rdf-syntax-ns"
rdfs:"http://www.w3.org/2000/01/rdf-schema"
protons:"http://proton.semanticweb.org/2005/04/protons"
protonu:"http://proton.semanticweb.org/2005/04/protonu"
musing:"http://www.ontotext.com/2007/07/musing"


"protons:Entity":
    SameAlias()

"protont:Company":
      let parentCond = Super() \& 0.7
          sectorCond =
          SameAttribute(<protonu:activeInSector>)
          aliasCond = SimilarCompanyAliases() \& 0.9
      in parentCond \& sectorCond \& aliasCond \&

"musing:Company":
      StrictSameAttribute("joci:hasURL")|
      OrganizationLD("joci:alias") |
      OrganizationCombine() \&
      StrictSameAttribute("joci:hasPostal") \&
      StrictSameAttribute("joci:hasSector")
```

Class models are evaluated differently by the SDCE depending on which component calls the engine.

**Formula evaluation during Pre-filtering** This component retrieves all instances in the knowledge base that are possibly identical to the currently processed object. In this case the engine does not formally evaluate the class model/formula but composes a SeRQL query. The query embodies the model restrictions with concrete values from the currently processed object e.g. restriction on attributes values, etc.

For example retrieving instances similar to a company called "MARKS & SPENCER" according to the class model for "musing:Company" on Figure 3.3 will result in a SeRQL query on Figue 3.3

*Example of SeRQL query for a "musing:Company" class model*

```
  select DISTINCT
  V1
from
  {V1} <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
              {<http://ontotext.com/2007/07/musing#Company>};
       [<http://ontotext.com/2007/07/joci#hasURL> {V2}];
       [<http://www.w3.org/2000/01/rdf-schema#label> {V3}];
       [<http://www.w3.org/2000/01/rdf-schema#label> {V4}];
       [<http://ontotext.com/2007/07/joci#hasPostal> {V5}]
where
  (V2 = "http://www.marksandspencer.com") or
  ( (   ( (V3 like "*marks" IGNORE CASE )
            or (V3 like "*marks *" IGNORE CASE )
        ) and
        ( (V4 like "*spencer" IGNORE CASE )
            or (V4 like "*spencer *" IGNORE CASE )
        )
    )
    and (V5 = "W2 1NW")
   )
```

Once the query is prepared it is sent to the semantic repository and the retrieved objects are returned to the pre-filtering component.

**Formula evaluation during Evidence Collection** This component calculates the similarity between two objects based on their class model, which is expressed by a probabilistic logic formula. Instead of firm logical values ($\top$ and $\bot$) used the classical logic, SCDE class model formulas are evaluated based on fuzzy logic. This means that all predicates that the formula consists of are calculated independently and then combined resulting in a real number from 0 to 1.

In is important to note that each formula is evaluated for a pair of instances - the new coming entity and the matching candidate. The same is true also

for the predicates calculation. Predicates have access to all details of both instances, although they can also take instance attributes, properties and relations as attributes.

Formally, a predicate value is calculated according to its algorithm, which reflects the specificity of the predicate and its attributes. For example *OrganizationLD()* predicate on Figure 3.3 computes Levenshtein Distance between two names(e.g. preprocessing suffixes and abbreviations).

Once calculated the values of different predicates are combined according to the logical connectives in the corresponding formula. In this setting the the usual logical connectives are expressed as arithmetic expressions:

$$a \vee b \equiv a + b - ab \tag{3}$$

$$a \wedge b \equiv ab \tag{4}$$

$$\overline{a} \equiv 1 - a \tag{5}$$

$$a \Rightarrow b \equiv \overline{a} \vee b \equiv 1 - a + ab \tag{6}$$

The evaluation of each formula for each pair of instances present the probability these two to be references to one and the same object, therefore they are matched with the calculated probability value. The result is a real value from 0 to 1, where value 0 means that the given entities are totally different and value 1 means that they are absolutely equivalent. Any value between 0 and 1 mean that these entities are equivalent but only with a specific confidence. Sometimes the similarity measure between two entities is based on the similarity between two other entities connected to the original one supported by usage of square bracket operator in the formula.

**Class Models execution** There are two different ways of using the Class models by the SDCE depending on which component used the engine.

- **Pre-filtering** component finds those objects in the knowledge base that are possibly identical to the instance candidate and it uses SDCE to acquire them. The engine is able to compose a SeRQL query based on the input object and corresponding class model. Then it send the query to the semantic repository and returns the retrieved objects to the pre-filtering component.
- **Evidence Collection** component calculates the similarity between two objects based on their class model, which is expressed by a probabilistic logic formula. The result is a real value from 0 to 1, where value 0 means that the given entities are totally different and value 1 means that they are absolutely equivalent. Any value between 0 and 1 mean that these entities are equivalent but only with a specific confidence. Sometimes the similarity measure between two entities is based on the similarity between two other entities connected to the original one supported by usage of square bracket operator in the formula.

## 4 Evaluation

We have carried out two series of experiments. In a first experiment we have merged

In a second experiment we have used RDF templates statements. The process has targetted a set of xxx UK companies and attempted unification agains an already populated knowledge base.

```
<rdf template="company+executive">
<protont:Person xmlns:protont="http://musing.deri.at/ontologies/v0.6/protont/protont#"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
rdf:ID="705f9163-68a7-41e5-8021-18b3d1803436">
<protonu:hasAlias xmlns:protonu="http://musing.deri.at/ontologies/v0.6/protonu/protonu#"
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
George G. Weston
</protonu:hasAlias>
<protonu:hasPosition xmlns:protonu="http://musing.deri.at/ontologies/v0.6/protonu/protonu#"
rdf:resource="4fa2b37f-d4e0-464c-9163-2509ff5879b8" />
</protont:Person>
<protont:JobPosition xmlns:protont="http://musing.deri.at/ontologies/v0.6/protont/protont#"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
rdf:ID="4fa2b37f-d4e0-464c-9163-2509ff5879b8">
<protont:description rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
Chief Executive Officer
</protont:description>
<protont:holder rdf:resource="705f9163-68a7-41e5-8021-18b3d1803436" />
<protont:withinOrganization rdf:resource="355c3487-3408-4532-946f-dd0e2b64f501" />
</protont:JobPosition>
<protonu:Company xmlns:protonu="http://musing.deri.at/ontologies/v0.6/protonu/protonu#"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
rdf:ID="355c3487-3408-4532-946f-dd0e2b64f501">
<protonu:hasAlias rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
ASSOCIAT BRIT FOODS
</protonu:hasAlias>
<protonu:hasEmployee rdf:resource="705f9163-68a7-41e5-8021-18b3d1803436" />
</protonu:Company>
<!--  RDFTemplate, generated at 20080205-111054  from file company-templates.xml
template company+executive  -->
</rdf>
```

## 5 Identity Resolution for Company Information

**Fig. 3.** Single Vacancy extraction from a web page

We present the evaluation of a process of resolution and merging of information in the context of extraction of company information from multiple sources.

Here we present the evaluation of the identity resolution framework in the context of a job vacancy extraction task which is a multi-source extraction problem. It uses the Internet as a source – web-sites of companies, job-boards and recruitment agencies – where it is possible to rely on many sources to improve the extraction of the facts we search for. The proposed approach is to extract all

available facts (in the job vacancy domain) from any single document, and then to combine/merge them on several levels to retrieve the most accurate facts, while at the same time filtering out wrong and redundant information.

### 5.1 Vacancy Extraction

The algorithm takes web-pages one by one and processes them separately, extracting listed vacancies. At this preliminary stage each page is pre-processed and certain types of named entities are recognised and annotated with respect to the ontology. After that, the set of extracted vacancies is further analysed to detect duplicates and finally inserted into the knowledge base.

Vacancy facts are defined by templates, which slots should be filled by concept instances in our KB. The extraction task consist on the extraction of the following information from text: JobTitle; ReportingTo; Job_Category; Job_Location; Location; Job_Reference; Job_Type; Salary; End_Date; Start_Date and Person. The proposed values for these attributes are named entities recognised by our system. Hence the extracted facts are actually a compilation of the attribute values in accordance with the domain constraints (see Figure 3). In Table 1, we present an evaluation of the extraction performance by slot. Overall, we have obtained F-score 87,4% (Precision 83,1% and Recall 92,3%) for single vacancy extraction.

| Attribute | Precision | Recall | F-measure |
|---|---|---|---|
| JobTitle | 0.87 | 0.86 | 0.85 |
| ReportingTo | 0.99 | 0.99 | 0.99 |
| Job_Category | 0.99 | 0.97 | 0.96 |
| Job_Location | 0.98 | 0.65 | 0.66 |
| Location | 0.89 | 0.93 | 0.88 |
| Job_Reference | 0.98 | 0.89 | 0.89 |
| Job_Type | 1 | 0.99 | 0.99 |
| Salary | 0.97 | 0.87 | 0.88 |
| End_Date | 0.99 | 0.93 | 0.93 |
| Start_Date | 0.98 | 0.98 | 0.89 |
| Person | 0.87 | 0.94 | 0.83 |

**Table 1.** Evaluation of single attributes extraction

### 5.2 Identity Resolution for Vacancy Merging

Once the vacancies are extracted the system proceeds with identification of those that are unique. For this purpose, we define Vacancy semi-equivalence is defined as follows: (i) equivalent "Vacancy Title" attribute values, or if one is a substring of the other, and (ii) the values of the rest of their attributes are semantically compatible according to the knowledge base, i.e. the two compared instances are connected with certain types of relation that is semantically consistent.

An example for such a relation is *subRegionOf* and we say that "*locatedIn* Wales" is comparable to "*locatedIn* UK", since "Wales" is a *subRegionOf* of "UK". What we achieve as a result of merging two vacancies is a new vacancy composed out of the most specific values among the two proposed values for each and for every attribute. All attribute values presented only in one of the merged facts are also taken. A very simple diagram on Figure 4 presents the choice of most specific values for *"Vacancy Title"* and *"Vacancy Location"* attributes presented as KB relations.

**Fig. 4.** Example of consolidation of two Vacancy facts

The motivation for the merging is the fact that one and the same vacancy is often promoted several times on a single (company) web-site. It starts from a list of vacant positions, followed by a very short description or a separate page with a detailed description of full vacancy details. The identity resolution is supported by the fact that all extracted position are offered in one and the same organisation. All this information gives us a chance to check the extracted facts and to collect all the available information provided by the employer when it is distributed on several pages.

Once having reliable single page IE results we investigate the redundancy phenomena. We took a sample of about 3k web sites and semi-automatically compared the extracted vacancies. Our experiment showed that about two thirds of the company web-sites have redundant job advertising. Moreover, the consolidation successfully reduces the number of facts to about 55% of the single page extracted results (see Table 2). The formal manual evaluation of the vacancy merging accuracy is given on Table 3.

| Statistics | |
|---|---|
| web-sites with extracted *Vacancies* | 2,922 |
| web-sites with redundancy | 2,171 |
| *Vacancies* before merging | 29,963 |
| *Vacancies* after merging | 16,592 |

**Table 2.** Redundancy Statistics

| Precision | Recall | F-measure |
|---|---|---|
| 0.82 | 0.89 | 0.85 |

**Table 3.** Evaluation of Intra-site vacancy merging

## 6 Related Work

Previous experiments in multi-source information extraction have been taken mainly in the area of Text Summarization, Databases and Co-reference Analysis. Bilenko and Mooney [**?**] present a framework for duplicate detection using trainable measure of textual similarity (a learnable text distance function). Comprehensive survey about different methods used for de-duplication in database field is given by [**?**]. However all the presented approaches are based on the string content of the corresponding field and hardly use even the fields' interdependence.

A notable aspect of using semantics for matching knowledge representation structures is presented by [**?**]. The authors define Match as an operator that takes two graph-line structures and produces mappings among the nodes that correspond semantically to each other. However, the processing is based mainly on the node labels, even if their comparison is based on WordNet [**?**] and the graph structure is restricted to a tree.

The IdRF proposed knowledge representation – ontologies – are already used for approaching the identity resolution problem. [**?**] present the advantages of semantically enhanced annotation for resolving co-references from different sources. Another example of using ontologies in this domain is the innovative work of [**?**] for extending standardised ontology description languages to unable approximation of instances. The authors introduce new "Rough Description Language" to represent and reason about similarity of instances.

From the natural language processing point of view, identity resolution has been addressed as a cross-document coreference task restricted to the problem of person coreference. Bagga and Baldwin [**?**,**?**] used the vector space model together with summarization techniques to tackle the cross-document coreference problem. They use a Vector Space Model Disambiguation module and compute similarities between personal summaries (sentences extracted) for each pair of documents. Summaries having similarity above a certain threshold are considered to be about the same entity. Mann and Yarowsky [**?**] use semantic information that is extracted from documents to inform a hierarchical agglomerative clustering algorithm. Semantic information here refers to factual information about a person such as the date of birth, professional career or education. Phan et al. [**?**] follow Mann and Yarowsky in their use of a kind of biographical information about a person. They use a machine learning algorithm to classify sentences according to particular information types. They compare information in automatically constructed person profiles by taking into account the type of the information. Entity identification is often addressed as author's name disambiguation in context of bibliographical records. In this context, Aswani el al. [**?**] base their approach on web searches while looking for the author home pages, as well as papers' titles and abstracts. They mine information from the Web for authors including full name, personal page, and co-citation information to compute the similarity between two person names. Similarity is based on a formula which combines numeric features with appropriate weights experimentally obtained. Finally, Saggion [**?**], studies the effect of different document contexts (e.g. full

document, summary) and term representations (e.g. words, named entities) for entity clustering. An approach which uses named entities of type organisation to disambiguate person names proved to be very competitive.

## 7 Conclusions and Future Work

We have presented a general framework for identity resolution which can be adapted to different ontology-based information extraction and ontology-population applications. We have also demonstrated and evaluated the application of the framework in the context of an ontology-based information extraction system. We are currently working on merging vacancies as well as organisations from sources different to corporate websites, e.g. job-boards. The approach taken uses consequential resolving of organisations followed by vacancy merging. Our future work will look into adapting the framework in the context of ontology population for business intelligence applications in financial risk management and internationalisation where target entities (e.g. companies, persons, locations) are extracted from multiple reduntant sources requiring consolidation.

## Acknowledgements

This work was partially supported by the EU-funded projects MUSING (IST-2004-027097) and MediaCampaing (027413).