Natural Language Processing for a Semantic Enabled Resource Retrieval Scenario

Azzurra Ragone†, Stefano Coppi†, Tommaso Di Noia†, Eugenio Di Sciascio†, Francesco M. Donini‡

Politecnico di Bari†
Via Re David, 200, I-70125, Bari, Italy
{a.ragone, s.coppi, t.dinoia, disciascio}@poliba.it

Università della Tuscia‡
via San Carlo, 32, I-01100, Viterbo, Italy
donini@unitus.it

Abstract— **We present an extended approach to resource retrieval using Natural Language Processing techniques. The aim is to ease user interaction with semantic-based facilitators. The approach allows the automated mapping of the semantics of a natural Language sentence with respect to an ontology modeled using Description Logics. The approach is embedded in the MAMAS matchmaking framework and exploits non-standard inference services for Description Logics.**

Keywords: natural language processing, description logics, e-commerce, parser, semantic web

I. Introduction

Knowledge-based systems for matchmaking and resource retrieval –often based on Description Logics fomalisms (DLs)– have been recently proposed [1], [2]. DLs, in fact, are endowed of a formal semantics, allow for an open-world assumption. Incomplete information is admitted, and absence of information can be distinguished from negative information. Furthermore such languages allow to model constraints of structured descriptions as concepts, which share a common ontology. Reasoning services allow to classify and rank resources, and –in novel advanced systems– provide explanation hypothesis when a match between requests and available resources is not exact [3], [4].

Obviously such characteristics are well different in comparison to those typical of DBMS or textual information retrieval systems(IRS). In the first ones the queries are computed based on a rigid and preset scheme, through predicates that can be verified or not, and the result exclusively corresponds to what satisfies such predicates. There is not formal semantics, and a closed world assumption is present. There is no automatic possibility of composition of resources with the purpose to answer to a request. IRS do not have the rigidity of the scheme and the concept of ranking is introduced –determined through some function that calculates the similarity between query and documents– but the indexation and the retrieval are carried out based on terms, without any relationship with the semantics of the text. Effective search engines, such as Google, exploit heuristics with good results, such as page ranking based on incoming and outgoing links and click counting, but they are not able to use formal semantics of structured annotations determined based on domain ontologies. User interaction with such systems is typically carried out using pre-determined forms, which can become cumbersome and time-consuming. On the other hand DL systems do not require rigid form schemas in principle, but average users will find irritating the need to learn about ontologies and constructors to interact with them. In this paper we report on how we faced such issues in the framework of MAMAS demand/supply semantic-matchmaking service [2], devising an approach specifically aimed at translating demand / supply descriptions expressed in Natural Language (NL) into structured DL expressions, mapping in an automated way NL sentences with concepts and roles of a DL-based ontology. Distinguishing characteristics of our NL parser include the direct use of DLs to express the semantic meaning, without intermediate stages in First Order Logic Form or lambda calculus. This has been possible because of the strong contextualization of the approach, oriented to e-commerce advertisements, which possess an ontological pattern that expresses their semantics and affects grammar creation. We define two distinct grammar categories for nouns: NP, which represent goods, and DP, which describe goods. These two categories reflect the two main branches of the upper ontology used to describe advertisements. This choice allows to embed the problem domain into the parser grammar. Furthermore we designed the grammar separate in two levels. In this way we achieve more flexibility: the first level only depends on the ontology terminology while the second only on the particular DL to be used.

II. Basics and Background Work

Since the early days of terminological reasoners, DLs have been applied in semantic interpretation for natural language processing [5]. Semantic interpretation is the derivation process from the syntactic analysis of a sentence to its logical form – intended here as the representation of its context-dependent meaning. Typically, DLs have been used to encode in a knowledge base both syntactic and semantic elements needed to drive the semantic interpretation process. For a recent survey of NLP projects using DLs, see Chapter 15 in [6]. To make the paper self-contained we briefly revisit fundamentals of DLs [6]. The basic syntax elements are *concept* names, such as, `Graduate`, `Programmer`; *role* names, such as `hasDegree`, `hasWorkingExperience`; *individuals*, such as `Julia`, `Michael`. Concepts stand for sets of objects, and roles link objects in different concepts. Individuals are used for special named elements belonging to concepts. Basic elements can be combined using *constructors* to form concept and role *expressions*, and each DL is identified by the operators set it is endowed with. Every DL allows one to form a *conjunction* of concepts, usually denoted as $\sqcap$; some DL include also disjunction $\sqcup$ and complement $\neg$ to close concept expressions under boolean operations. Expressive DLs [6] are built on the simple $\mathcal{AL}$ (Attributive Language) adding constructs in order to represent more expressive concepts. Allowed constructs in $\mathcal{AL}$ are: $\top$ *universal concept* (all the objects in the domain); $\bot$ *bottom concept* (the empty set); A *atomic concepts* (all the objects belonging to the set represented by A); $\neg A$ *atomic negation* (all the objects not belonging to the set represented by A); $C \sqcap D$ *intersection* (the objects belonging both to C and D); $\forall R.C$ *universal restriction* (all the objects participating to the R relation whose range are all the objects belonging to C); $\exists R$ *unqualified existential restriction* (there exists at least one object participating in the relation R). Expressions are given a semantics by defining the interpretation function over each construct. Concept conjunction is interpreted as set intersection: $(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$, and also the other boolean connectives $\sqcup$ and $\neg$, when present, are given the usual set-theoretic interpretation of union and complement. The interpretation of constructs involving quantification on roles needs to make domain elements explicit: for example, $(\forall R.C)^{\mathcal{I}} = \{d_1 \in \Delta \mid \forall d_2 \in \Delta : (d_1, d_2) \in R^{\mathcal{I}} \rightarrow d_2 \in C^{\mathcal{I}}\}$. Concept expressions can be used in *inclusion assertions*, and *definitions*, which impose restrictions on possible interpretations according to the knowledge elicited for a given domain.

Adding new constructors to $\mathcal{AL}$ increases DL languages expressiveness, but may also make inference services intractable [7]. The allowed operators in a DL based an $\mathcal{AL}$ are indicated by a capital letter. For instance, $\mathcal{ALN}$ is a $\mathcal{AL}$ endowed with unqualified number restriction *i.e.*, $(\geq\ n\ R)$, $(\leq\ n\ R)$, $(=\ n\ R)$ (respectively the minimum, the maximum and the exact number of objects participating in the relation R); $\mathcal{ALC}$ allows full negation; in $\mathcal{ALE}$ there can be used the qualified existential restriction; in $\mathcal{ALEN}$ both existential and unqualified number restriction are defined and so on. Here we refer mainly to an $\mathcal{ALN}$ DL, which can be mapped in a subset of OWL-DL [8]. Ontologies using Description Logics can be easily modeled using languages for the Semantic Web. The strong relations between Description Logics and the above introduced languages for the Semantic Web [9] is also evident in the definition of the OWL language, particularly in its sub-language OWL-DL. Besides the computational tractability of its related algorithms, further motivations for choosing the $\mathcal{ALN}$DL are explained in Section V In the rest of the paper we will use DL syntax instead of OWL-DL syntax, because of its conciseness.

III. Inference Services in Description Logics

A. Standard Inference Services

DL-based systems usually provide two basic reasoning services:

1) *Concept Satisfiability*: given a TBox $\mathcal{T}$ and a concept C, does there exist at least one model of T assigning a non-empty extension to C?
2) *Subsumption*: given a TBox $\mathcal{T}$ and two concepts C and D, is C more general than D in any model of $\mathcal{T}$?

Although these two basic services are very useful , there are cases where there is the need to go beyond subsumption and satisfiability.

B. Non-Standard Inference Services

In [10] Concept Abduction Problem (CAP) is introduced as a new non-standard inference problem for Description Logics.

Definition 1: Let C, D, be two concepts in a Description Logic $\mathcal{L}$, and $\mathcal{T}$ be a set of axioms, where both C and D are satisfiable in $\mathcal{T}$. A *Concept Abduction Problem* (CAP), denoted as

$\langle \mathcal{L}, S, D, \mathcal{T} \rangle$, is finding a concept E such that $\mathcal{T} \not\models C \sqcap E \equiv \bot$, and $\mathcal{T} \models C \sqcap E \sqsubseteq D$.
In [10] also minimality criteria for E and the polynomial algorithm, *abduce*, to find solutions which are irreducible, for the $\mathcal{ALN}$ DL, have been proposed. Given a CAP, if E is a conjunction of concepts and no sub-conjunction of concepts in E is a solution to the CAP, then E **is an irreducible solution**. The solution to a CAP can be read as *why does not $C \sqcap D$ hold?, what is a possible* ***explanation*** *with respect to the ontology $\mathcal{T}$?* In other words E is what is expressed, explicitly or implicitly (that is, entailed because of the ontology), in D and is not present in C, or also *which part of D is not covered by C*.

In [2] the *rankPotential* algorithm is presented such that, given a set of $\mathcal{ALN}$axioms $\mathcal{T}$ and two $\mathcal{ALN}$concepts C and D both satisfiable in $\mathcal{T}$, it computes a *semantic distance* of C from D with respect to the ontology $\mathcal{T}$. Notice that we write *the distance of D from C* rather then *the distance between C and D* because of the non-symmetric behavior of *rankPotential* (see [2] for further details). With the aid of *rankPotential* it is also possible to compute a complex concept depth with respect to the taxonomy represented by the axioms set $\mathcal{T}$. In fact, if $C \equiv \top$ then *rankPotential(C,D)* = *rankPotential($\top$,D)* represents the distance of D from $\top$, *i.e.*, the most generic concept in the ontology. Such distance is not simply the depth of a node in a tree at least for two main reasons:
1. An $\mathcal{ALN}$ontology, typically, is not a simple terms taxonomy tree, *i.e.*, it not contains only IS-A relations between two atomic concepts, and can be better represented as a labeled oriented graph.
2. An $\mathcal{ALN}$complex concept is the conjunction of atomic concepts and role expressions.

The value returned by *rankPotential($\top$,D)* here represents how specific is a complex concept expression D with respect to an ontology $\mathcal{T}$.

IV. Description Logics Non-Standard Reasoning for Resource Retrieval Scenario

In a marketplace scenario subsumption and satisfiability can be efficiently used to discover semantic (un)satisfiability of a supply with respect to a demand. For instance, consider you are requesting for DEMAND and in your marketplace there is SUPPLY; if the SUPPLY $\sqcap$ DEMAND $\equiv \bot$ relation ensues, then SUPPLY is not compatible with DEMAND. On the other hand, if SUPPLY $\sqsubseteq$ DEMAND then supply SUPPLY completely satisfies DEMAND. In a marketplace scenario, the latter relation represents the (unfrequent) ideal situation when an advertised supply completely satisfies the user request. If the relation is not valid and SUPPLY $\not\sqsubseteq$ DEMAND, it should be feasible to know what is needed in SUPPLY in order to satisfy DEMAND. For such purpose both *abduce*[10] and *rankPotential* [2] can be used. Using the former an irreducible explanation, E, is computed on why the subsumption relation does not hold. With the latter, numerical evaluations of the distance of DEMAND from SUPPLY and the depth of both DEMAND and E can be evaluated, with respect in a semantic graph represented by the ontology.

Computing the above three distinct values (distance, DEMAND depth and E depth) helps the user in evaluating non-exact matches between a demand and a supply. The distance is a preliminary filter to rank appealing supplies with respect to a demand. The less distance the better. When the computed distance is the same for two or more supplies with respect to the same demand, the ratio between the depth of E and the DEMAND's one assesses how specific is the part of the demand on whom nothing is specified in SUPPLY. In this case the more specific the better. The observation behind the previous idea is that the user could retract on requested information unspecified/missing in the SUPPLY. Hence, the more specific the missing information is, the more the user can relax and underspecify it.

V. A Grammar for Parsing Advertisements

In this section we show how to model a NL sentence as a DL concept expression in order to perform a semantic based matchmaking process between the demand and the supply, which exploits the non-standard inference services presented in Section III-B.

To establish rules and models for translation, we started analyzing several advertisements related to different commerce domain *e.g.*, consumer electronics components, real estate services, job postings. As we expected, we noticed that advertisements present almost always, regardless of the domain, a characteristic structure and are strongly contextualized. Furthermore the lexicon often uses some jargon and is a finite and limited set of terms. With reference to the structure, there is always the good(s)/service to be bought/sold and related characteristics. Each good in the domain refers to a single concept in the knowledge domain but can be represented using different expressions, which are semantically equivalent. The same can be said for good characteristics. Hence, in each sentence there are at least two main lexical category: the good and its description.

From a DL point of view, generic advertisement can be brought back to the following form:

$$C_1 \sqcap C_2 \sqcap ...C_n \sqcap \forall r_1.D_1 \sqcap \forall r_2.D_2 \sqcap ...\forall r_m.D_m$$

where generally C_i are the concepts related to the goods, and $\forall r_j.D_j$ to the goods description. This pattern can be also used as a guideline to model the task ontology for the specific marketplace. Atomic concepts representing a good are modeled as subconcepts of a generic `Goods` concept. Notice that at least an $\mathcal{ALN}$ DL is needed to model correctly a marketplace, in order to deal with concept taxonomy, disjoint groups, role restrictions ($\mathcal{AL}$), and particularly number restriction ($\mathcal{N}$) to represent quantity.

The sentence structure led us to investigate techniques similar to *Semantic Grammars* [11] ones, where the lexical categories are based on the semantic meaning. We created two basic lexical categories: Fundamental Names (FN), denoting noun phrases representing goods, and simple Names (N), denoting noun phrases describing goods. This distinction is useful during grammar rules composition (see V-A.2) because it allows to determine if a sentence is acceptable or not in our scenario. It must contain at least a constituent of category FN, otherwise it means there are no goods to look for. Since the idea was to bind the grammar to the reference DL ontology, we thought to enforce the relationship using features identifying the role of lexical categories within the ontology itself. Inspired by the use of a **TYPE** feature in a *Template Matching* [11] approach, we created five different features, respectively for concept names (`concept`), role names (`role`), operators (`op`), role restrictions (`rr`) and an auxiliary feature (`aux`), whose value is strictly related to the terminology used in the ontology. Using such features it is possible both to align the lexicon with the terms in the ontology and to obtain a limited number of rules associating a semantic meaning to the constituents.

A. Lexicon and Grammars

With the aim of building reusable elements to be easily adapted for different marketplaces and ontologies, we separated information related to the terminology, the lexical category of the terms, and the expressiveness of the DL used to model the ontology. The idea is to minimize changes and possibly to reuse both the lexical and the semantic information. In fact the parsing process is conceived in two stages, each one using a different (kind of) grammar. Using the first grammar, terms in the NL sentence are strictly related both to the terminology used in the ontology –atomic concept names and role names– and to the logical operators. With the Level 1 Grammar a parser is able to bind set of words to the correspondent element in the ontology. The Level 2 grammar uses the intermediate result produced during the Level 1 phase to build the logical form of the sentence with respect to a good/description model. In this parsing phase logical operators and quantifiers allowed by the DL used to built the ontology are used to link the basic elements. This subdivision allows more flexibility. Adapting the grammar to a new ontology (based on the same DL) requires major changes only in the Level 1 grammar, in which concept and role names appear, in order to remap the new Lexicon to the terminology used in the ontology. On the other hand if the adopted DL is changed, *e.g.*, from a $\mathcal{ALN}$ DL to a $\mathcal{ALEN}$ DL [6], major changes are requested only for Level 2 rules.

In the following we show how the logical form of the sentence is built with the aid of some examples, conceived with reference to the toy ontology in Fig. 1.

...

$\texttt{Graduate} \equiv \exists \texttt{hasDegree} \sqcap \exists \texttt{hasTechnicalPrerequisites}$

$\texttt{Engineer} \equiv \texttt{Graduate} \sqcap \forall \texttt{hasDegree.Engineering}$

$\texttt{CSEngineer} \equiv \texttt{Engineer} \sqcap$

$\forall \texttt{hasTechnicalPrerequisites.ComputerProgramming}$

$\texttt{Programmer} \sqsubseteq \texttt{Person} \sqcap \exists \texttt{hasTechnicalPrerequisites} \sqcap$

$\forall \texttt{hasTechnicalPrerequisites.ComputerProgramming}$

$\texttt{TechnicalPrerequisites} \sqsubseteq \texttt{Prerequisites}$

$\texttt{ComputerProgramming} \sqsubseteq \texttt{TechnicalPrerequisites}$

$\texttt{OOP} \sqsubseteq \texttt{ComputerProgramming}$

$\texttt{DistributedProgramming} \sqsubseteq \texttt{ComputerProgramming}$

$\texttt{Corba} \sqsubseteq \texttt{DistributedProgramming}$

$\texttt{Corba} \sqsubseteq \texttt{OOP}$

$\texttt{WebService} \sqsubseteq \texttt{DistributedProgramming}$

$\texttt{Corba} \sqsubseteq \neg\texttt{WebService}$

$\texttt{NorthAmerica} \sqsubseteq \texttt{America}$

$\texttt{Usa} \sqsubseteq \texttt{NorthAmerica}$

$\texttt{California} \sqsubseteq \texttt{Usa}$

...

Fig. 1. The toy ontology used as reference in the paper

1) Lexicon: First of all, we point out that for our purpose, a morphological analysis can be avoided without loss of generality. In the lexicon, each term is endowed with the following features:

- `cat` represents the lexical category of the single word, *i.e.*, FN (noun indicating goods), N (noun describing goods), V (verb), ADJ

1) NP[c,-,-,-] → FN[c,-,-,-]
2) DPF[c2,r1,-,-] → V[-,r1,-,-] N[c2,-,-,-]
3) DP[-,r2,c1,-] → ADJN[c1,-,-,-] N[-,r2,-,-]
4) DP[-,concat(r1,r2),-,-] → V[-,r1=hasWorking,-,-] N[-,r2=Experience,-,-]
5) N[concat(c1,c2),-,-,-] → N[c1=North,-,-,-] N[c2=America,-,-,-]
6) DP[-,-,-,atLeast] → PREP[c1=at,-,-,-] ADV[c2=least,-,-,-]

Fig. 2. Example Level 1 Grammar Rules

1) DPE[(And (atLeast x r) (atMost x r))] → DP[-,r,x,-]
2) DPL[(atLeast x2 r2)] → DP[-,-,-,a1=atLeast] DP[-,r2,x2,-]
3) DPA[(all r c)] → DPF[c,r,-,-]
4) DPA[(all r2 c1)] → DPE[c1,-,-,-] DP[-,r2,-,-]
5) DPA[(all r2 c1)] → DPL[c1,-,-,-] DP[-,r2,-,-]
6) DPA[c1 c2] → DPA[c1,-,-,-] DPA[c2,-,-,-]
7) S[(And c1 c2)] → NP[c1,-,-,-] DPA[c2,-,-,-]

Fig. 3. Example Level 2 Grammar Rules

(adjective), ADJN (numerical adjective), ADV (adverb), ART (article), CONJ (conjunction), PREP (preposition).

- `concept,role` represent, respectively, the corresponding atomic concept, role in the ontology.
- `op` represents the corresponding logical operator in DL.
- `sw`, is set `true` if the term is a *stopword.*
- `aux` is an auxiliary field for a further customization of the grammars.

2) Level 1 Grammar: Actually, the mapping between the terms in the NL sentence and the ones in the ontology is not in a one to one relationship. There is the need to relate words set to the same concept or role within the ontology. In Fig. 2 a simple grammar is reported to deal with sentences related to our reference skill management domain (see Fig. 1).

1) maps nouns FN to constituents NP which can contain more than one noun.
2) maps a role r1 to its corresponding filler, the concept c2 creating a constituent DPF.
3) since number restriction are needed in e-commerce scenarios, as good descriptions, we allow to introduce them in this grammar. Rule 3) creates a new DP constituent linking the role $years$ to its numerical restriction, *e.g.*, $(= \ 2 \ years)$.

4)5) deals with elements in the ontology represented by two or more words in the sentence. In particular, Rule 5) creates a consituent N, having as concept the conjunction of the words “North” and “America”.

6) create a constituent DP, with the feature *aux=atLeast*. This auxiliary feature, in this case, represents the type of role restriction.

3) Level 2 Grammar: This grammar binds the sentence to the expressiveness of the DL chosen to model the ontology. The purpose of Level 2 rules is to put together single concepts and roles of the ontology, to form an expression in DL representing the logical model of the sentence, reflecting the structure of the good/description ontological pattern. With respect to the rules in Fig. 3 we obtain:

1) introduces the DL operator $=$ In fact rule 1) states that if there is a constituent DP, *e.g.*, with `role="years"` and `role restriction="2"`, a new DPE (a descriptive constituent containing an exactly restriction) is created with `concept` containing the DL expression: $(\geq \ 2 \ years) \sqcap (\leq \ 2 \ years)$.
2) introduces the DL operator $\geq$ In fact states that if there is a DP with the `aux="atLeast"`, followed by a DP with `role="years"` and `role restriction="2"`, a new DPL (a descriptive constituent containing an at-least restriction) is created with `concept` containing the DL expression: $(\geq \ 2 \ years)$.
3) introduces the DL operators $\forall$. This rule states that if there is a constituent DPF, *e.g.*, with `role="livingIn"` and `concept="California"`, a new DPA (a descriptive constituent containing the $\forall$ operator) is created with `concept` containing the DL expression: $\forall livingIn.California$.

4)5)6) are useful to compose contiguous constituents of the same type.

7) states that a sentence is composed by a constituent NP representing the good of the advertisement, followed by a descriptive constituent DPA .

B. Ambiguity Resolution Through Filtering

After the parsing process, more then one DL logical expression corresponding to the NL sentence can be produced. A solution set analysis is used to filter the number of expressions to only one.

Removing unsatisfiable descriptions. Descriptions unsatisfiable with respect to the ontology are filtered out.

Ontological pattern matching. A more selective filter is the one checking if the DL descriptions match a given ontological pattern. In the marketplace scenario it is verified if the concept expressions keep the good/description structure via a subsumption check with a DL expression representing such structure. After the application of the previous filters, there could yet be more than one DL expression $D_1, D_2, ..., D_n$ associated to the sentence. In order both to avoid the same sentence being described with logical formulas inconsistent with each other and to put together all the information extracted from the NL sentence, we model the final translation as the conjunction of all the translation remained after previous filtering. Doing so, if two result descriptions, D_i,D_j, model information incompatible with each other, *i.e.*, $D_i \sqcap D_j \equiv \bot$, then an error occurs stating that it is not possible to find a unique semantic model of the sentence. On the other side, due to grammar rule choosing during the parsing process, it happens that some information is not modeled in all the result expressions. Hence, with the conjunction of all the resulting DL expressions we join all the information that can be caught by a parser.

VI. Example

As an example consider a skill-management scenario for personnel recruitment. Here your request is:
"Looking for Computer Science Engineer, specialized in CORBA, with at least 2 years working experience, living in California".
Its DL representation, derived from the parsing process using the grammars in Figure 2 and Figure 3, is:
$D = \texttt{CSEngineer} \sqcap \forall\texttt{livingIn}.\texttt{California} \sqcap$
$\forall\texttt{hasTechnicalPrerequisites}.\texttt{Corba} \sqcap$
$\forall\texttt{hasWorkingExperience}.(\geq\ 2\ \texttt{years})$
Now, imagine two available candidates for your advertisement.

Julia : "Computer Science Engineer, living in North America, with 3 years working experience".
Richard : "Programmer, specialized in CORBA, with 2 years working experience, living in California".

The DL model associated to the previous NL sentences are respectively:

Julia: $\texttt{CSEngineer} \sqcap \forall\texttt{livingIn}.\texttt{NorthAmerica} \sqcap$
$\forall\texttt{hasWorkingExperience}.(=\ 3\ \texttt{years})$

Richard: $\texttt{Programmer} \sqcap \forall\texttt{livingIn}.\texttt{California} \sqcap$
$\forall\texttt{hasTechnicalPrerequisites}.\texttt{Corba} \sqcap$
$\forall\texttt{hasWorkingExperience}.(=\ 2\ \texttt{years})$

Computing the semantic distance of both *Julia* and *Richard* from D and the mismatch explanation H, you obtain:

rankPotential(Julia,D) = 5
$H_{Julia} = \forall\texttt{livingIn}.\texttt{California} \sqcap$
$\sqcap\forall\texttt{hasTechnicalPrerequisites}.\texttt{Corba}$
$|H_{Julia}| = 10$

rankPotential(Richard,D) = 5
$H_{Richard} = \texttt{CSEngineer}$
$|H_{Richard}| = 12$

In this case we can observe that both *Julia* and *Richard* have the same semantic distance (result of rankPotential) from D. Using only the semantic distance as ordering criterion, there is no difference between *Julia* and *Richard*. Using, in a second step, the information related to H_{Julia} and $H_{Richard}$ it's possible to resolve similar cases. In fact, looking at the depth of H, we can observe that $|H_{Richard}| > |H_{Julia}|$. The previous relation means that what is not covered in the request by Richard is more specific than what is not coverd by Julia. Since the more specific the missing information is, the more the user can relax and underspecify it, we choose the supply having the greatest depth of H; in this case is *Richard*.

VII. Experiments and Results Discussion

In order to validate both the NLP approach and the matchmaking one, we carried out experiments on human users in a job recruitement scenario. In such experiment, the user formulated her/his request and he was asked to order a set of eight advertisements (supplies) on the basis of the request. After the user ranking, a parser analyzed both the request and the supplies extracting their DL model and a matchmaker computed their semantic similarity.

Initially, the matchmaker ranked supplies based only on the result returned by the *rankPotential* algorithm. After an analysis of the mismatch results between the user ranking and the matchmaker one, we argued that the user result depends also on the depth of E returned by *abduce*. That is, the user choice is based not only on the semantic distance between her/his demand and the supplies, but also on *how specific* is, with respect to the ontology, the uncovered part of the request.

Acknowledgements

This work was partially supported by projects CNOSSO, MS3DI, PITAGORA.

References

[1] L. Li and I. Horrocks, "A Software Framework for Matchmaking Based on Semantic Web Technology," *International Journal of Electronic Commerce*, vol. 8, no. 4, 2004.

[2] T. Di Noia, E. Di Sciascio, F. Donini, and M. Mongiello, "A system for principled Matchmaking in an electronic marketplace," *International Journal of Electronic Commerce*, vol. 8, no. 4, pp. 9–37, 2004.

[3] S. Colucci, T. Di Noia, E. Di Sciascio, F. Donini, and M. Mongiello, "Concept Abduction and Contraction for Semantic-based Discovery of Matches and Negotiation Spaces in an E-Marketplace," *Electronic Commerce Research and Applications*, vol. 4, no. 3, 2005.

[4] S. C. T. Di Noia, E. Di Sciascio, F. Donini, and A. Ragone, "Knowledge Elicitation for Query Refinement in a Semantic-Enabled E-Marketplace," in *Proceedings of the 7th Intl. Conf. on Electronic Commerce, ICEC'05*, 2005, to appear.

[5] R.Brachman, R.Bobrow, P.Cohen, J.Klovstad, B.Webber, and W.Woods, "Research in natural language understanding, annual report," Bolt Beranek and Newman, Tech. Rep. 4274, 1979.

[6] F. Baader, D. Calvanese, D. Mc Guinness, D. Nardi, and P. Patel-Schneider, Eds., *The Description Logic Handbook*. Cambridge University Press, 2002.

[7] R. Brachman and H. Levesque, "The tractability of subsumption in frame-based description languages," in *Proc. AAAI-84*, 1984, pp. 34–37.

[8] T. Di Noia, E. Di Sciascio, and F. Donini, "Extending Semantic-Based Matchmaking via Concept Abduction and Contraction," in *Proc. EKAW 2004*, 2004, pp. 307–320.

[9] F. Baader, I. Horrocks, and U. Sattler, "Description logics as ontology languages for the semantic web," in *Festschrift in honor of Jörg Siekmann*, 2003.

[10] T. Di Noia, E. Di Sciascio, F. Donini, and M. Mongiello, "Abductive matchmaking using description logics," in *Proc. IJCAI-03*, 2003, pp. 337–342.

[11] J. Allen, *Natural Language Understanding (2nd ed.)*. The Benjamin Cummings Publishing Company Inc., 1999.