Semantic Matchmaking and Ranking: Beyond Deduction in Retrieval Scenarios

Tommaso Di Noia

SisInf Lab – DEE – Politecnico di Bari – Via E. Orabona, 4, 70125 Bari, Italy
t.dinoia@poliba.it

Extended Abstract

Matchmaking can be basically seen as the process of computing a ranked list of resources with respect to a given query. *Semantic matchmaking* can be hence described as the process of computing such ordered list also taking into account the semantics of resources description and of the query, provided with reference to a logic theory (an ontology, a set of rules, etc.) [3]. A matchmaking step is fundamental in a number of retrieval scenarios spanning from (Web) service discovery and composition to e-commerce transactions up to recruitment in human resource management for task assignment, just to cite a few of them. Also in interactive exploratory tasks, matchmaking and ranking play a fundamental role in the selection of relevant resources to be presented to the user and, in case, further explored. In all the above mentioned frameworks, the user query may contain only hard (strict) requirements or may represent also her preferences. We will see how to handle both cases while computing the ranked list of most promising resources.

In semantic-based retrieval scenarios we have a logic theory represented, for example, as a set of ontological axioms that logically constrain the query and the description of a resource. If we consider both the query q and the description d expressed as Description Logics (DL)[1] statements (the logic language behind the semantics of `OWL`) and an ontology O we may check if the information encoded in d implies the one encoded in q or we may verify if d contains the description of some characteristics that are conflicting with the query. Let us consider the following simple example. Suppose we have a query q and four descriptions d_1, d_2, d_3 and d_4 such that the following relations hold[2]:

$$d_1 \sqcap q \not\sqsubseteq_O \bot \qquad d_2 \sqcap q \not\sqsubseteq_O \bot \qquad d_3 \sqcap q \sqsubseteq_O \bot \qquad d_4 \sqsubseteq_O q$$

By looking at the above relations we may say that, for sure, d_4 is the best choice for q since it *fully* satisfies the requirements expressed by the query. On the other hand, if we rely on deductive reasoning tasks, regarding d_1, d_2 and d_3 we may only say that the first two are compatible with q and the third one is not

[1] In the following we will use DL notation to present our framework but the approaches can be easily adapted to other logic formalisms.

[2] We use the notation $A \sqsubseteq_O B$ to represent $O \models A \sqsubseteq B$.

compatible with q since it contains some elements that make $q \sqcap d_3$ unsatisfiable. If we adopt an Open World Assumption (which is reasonable in many Web scenarios), we may say that d_1 and d_2 *potentially* satisfy q. After all, we do not have information on what is underspecified in both d_1 and d_2. Maybe, the one who wrote the description (imagine an advertisement on eBay) just did not care about some information to be explicitly represented. On the other hand, even though d_3 is conflicting with q, it may still contain some characteristics that have been expressed in q. In other words, d_3 *partially* satisfies q. The main questions here are: How do we rank d_1, d_2 and d_3 with respect to q? Is the classification in *potential* and *partial* matches[3] enough for ranking purposes? May we compute a similarity degree of d_1, d_2 and d_3 with respect to q based on their formal semantics [7]?

Our proposal is to use two non standard reasoning tasks, namely Concept Abduction and Concept Contraction [5], to compute an explanation on "the reason why" d does not *fully* satisfy q and then compute a score based on this explanation [4,6]. In a few words, we want to know the reason why $d \sqsubseteq_O q$ does not hold.

Concept Abduction. As for the Propositional case, the main aim of this task is to compute a (minimal) hypothesis on what is underspecified in d in order to *fully* satisfy q. More formally, given d, q and O such that $d \sqcap q \not\sqsubseteq_O \bot$, we say that h is a solution to a **C**oncept **A**bduction **P**roblem $\langle q, d, O\rangle^{CAP}$ if the following two conditions hold: $d \sqcap h \not\sqsubseteq_O \bot$ and $d \sqcap h \sqsubseteq_O q$. By solving a CAP on all those descriptions that *potentially* satisfy q, we have an explanation for *non-full* match. At this point, we may define a scoring function that assigns a weight to h, given q, d and O. Actually, the above formulation shows its limits with expressive languages and a more sophisticated definition is needed [1,8,2].

Concept Contraction. In case of partial match, we are interested in computing which part of the query q is conflicting with d and suggest to the user to revise and relax the query if she is interested in d. More formally, given d,q and O such that $d \sqcap q \sqsubseteq_O \bot$ we say that $\langle g, k\rangle$ is a solution to a **C**oncept **C**ontraction **P**roblem $\langle q, d, O\rangle^{CCP}$ if the following two conditions hold: $g \sqcap k \equiv_O q$ and $k \sqcap d \not\sqsubseteq_O \bot$. In a few words, g represents what has to be given up from q while k what has to be kept in order to *potentially* match q. Also in this case we may define a scoring function that assigns a weight to g and k, given q, d and O. Both for h and $\langle g, k\rangle$, some minimality criteria need to be defined in order to avoid trivial solutions.

Looking at the previous two reasoning tasks we observe that they could be used to move from a *partial* match to a *potential* match and from a *potential* match to a *full* match by solving a CCP and then a CAP. Indeed, we have the following matchmaking "path":

$$\underset{\textbf{\textit{partial}}}{q \sqcap d \sqsubseteq_O \bot} \xrightarrow{\langle g,k\rangle = solve\ \langle q,d,O\rangle^{CCP}} \underset{\textbf{\textit{potential}}}{k \sqcap d \not\sqsubseteq_O \bot} \xrightarrow{h = solve\ \langle k,d,O\rangle^{CAP}} \underset{\textbf{\textit{full}}}{d \sqcap h \sqsubseteq_O k}$$

[3] These two match classes are also known as *intersection* and *disjoint* respectively [10].

We notice two important aspects related to h, g and k:

- they represent an **explanation** on the reason why d is not a *full* match for q. This is a very relevant aspect of modern retrieval systems and it is a very hot topic, for example, in the field of Recommender Systems [16].
- they can be used as a metric to compute a **semantic distance** of d from q. Hence we can use them all to evaluate the score we will use to rank a set of descriptions with respect to the query.

So far, we have represented the query as a single formula without taking into account user preferences. In fact, it may happen that a user is more satisfied by the retrieved d depending on which part of her query q is satisfied: the user may assign a different *utility* value to different parts of q. In these situations we need to borrow some notions and ideas from Utility Theory [9] to perform an efficient retrieval task. To model user preferences and their associated utility we represent them as weighted formulas. A preference is then formulated as $P = \langle p, v\rangle$, where p is a logic formula and v is a numerical value representing the utility gained by the user when d satisfies p. Given a set of preferences $\mathcal{P} = \{\langle p_i, v_i\rangle\}$ and a resource description d we define a utility function, in its basic form, as

$$u'(\mathcal{P}, d) = \sum\{v_i \mid \text{where } d \text{ satisfies } p_i\}$$

Actually, we still need to define what "where d satisfies p_i" really means. The most intuitive way of modelling such relation is $d \sqsubseteq_O p_i$. Unfortunately, this formulation may lead to some counter-intuitive situations. Suppose to have $d = A_1 \sqcup A_2$, $P_1 = \langle A_1, v_1\rangle$ and $P_2 = \langle A_2, v_2\rangle$. In this case, neither $d \sqsubseteq A_1$ nor $d \sqsubseteq A_2$ and then $u'(\{P_1, P_2\}, d) = 0$. This is not correct since we know, by the formal semantics of $\sqcup$, that d will satisfy A_1 or A_2 or them both. Hence, the final utility will be at least $min(\{v_1, v_2\})$. In order to solve such problems we need a more sophisticated utility function that takes into account the models of the formula d. Given an interpretation $\mathcal{I}$ of d, *i.e.*, $d^{\mathcal{I}} \neq \emptyset$, we say that $\mathcal{I}$ is a minimal model of d if the value

$$u^{min}(\mathcal{P}, \mathcal{I}) = \sum\{v_i \mid \langle p_i, v_i\rangle \in \mathcal{P} \text{ and } \mathcal{I} \models p_i\}$$

is minimal. In this case, we call $u^{min}(\mathcal{P}, \mathcal{I})$ a **minimal utility value** associated to d with respect to $\mathcal{P}$. As we would expect, things become a little bit more complicated when we need to consider also an ontology [14,15].

In all the proposed examples we always had only one active user who formulated the query q (as a set of preferences). Nevertheless, we can model negotiating actors where both the user who expresses q and the one who describes d have an active role in the matchmaking and they both want to maximize their expected utility. Imagine the case of an online marketplace where the seller and the buyer have some preferences on the configuration of an item to sell and to buy respectively, and there is a mediator (in this case the marketplace itself) that tries to find an agreement that is mutually beneficial for both the buyer and the seller. We have successfully investigated such frameworks with particular reference to multi-issue *bilateral one-shot negotiation*, a special case of *bilateral matchmaking*.

Among all possible Pareto efficient [11] agreements we are obviously interested in those either maximizing the sum of utilities – maximum welfare – or maximizing their product – Nash-bargaining solution [12,13] and we show how to compute them.

References

1. S. Colucci, T. Di Noia, E. Di Sciascio, F.M. Donini, and M. Mongiello. A Uniform Tableaux-Based Method for Concept Abduction and Contraction in Description Logics. In *ECAI '04*, pp. 975–976, 2004.
2. S. Colucci, T. Di Noia, E. Di Sciascio, F. M. Donini, and A. Ragone. A unified framework for non-standard reasoning services in description logics. In *ECAI '10*, pp. 479–484, 2010.
3. S. Colucci, T. Di Noia, A. Pinto, A. Ragone, M. Ruta, and E. Tinelli. A non-monotonic approach to semantic matchmaking and request refinement in e-marketplaces. *IJEC*, 12(2):127–154, 2007.
4. T. Di Noia, E. Di Sciascio, and F.M. Donini. Extending Semantic-Based Matchmaking via Concept Abduction and Contraction. In *EKAW '04*, pp. 307–320, 2004.
5. T. Di Noia, E. Di Sciascio, and F.M. Donini. Semantic Matchmaking as Non-Monotonic Reasoning: A Description Logic Approach. *JAIR*, 29:269–307, 2007.
6. T. Di Noia, E. Di Sciascio, and F.M. Donini. Semantic matchmaking via non-monotonic reasoning: the mamas-tng matchmaking engine. *Communications of SIWN*, 5:67–72, 2008.
7. T. Di Noia, E. Di Sciascio, F.M. Donini, and M. Mongiello. A system for principled Matchmaking in an electronic marketplace. *IJEC*, 8(4):9–37, 2004.
8. T. Di Noia, E. Di Sciascio, and F. M. Donini. Computing information minimal match explanations for logic-based matchmaking. In *WI/IAT '09*, pp. 411–418, 2009.
9. P. C. Fishburn. Interdependence and additivity in multivariate, unidimensional expected utility theory. *International Economic Review*, 8:335–342, 1967.
10. L. Li and I. Horrocks. A Software Framework for Matchmaking Based on Semantic Web Technology. *IJEC*, 8(4):39–60, 2004.
11. R. B. Myerson and M. A. Satterthwaite. Efficient mechanisms for bilateral trading. *Journal of Economic Theory*, 29(2):265–281, 1983.
12. A. Ragone, T. Di Noia, E. Di Sciascio, and F.M. Donini. Logic-based automated multi-issue bilateral negotiation in peer-to-peer e-marketplaces. *J.AAMAS*, 16(3):249–270, 2008.
13. A. Ragone, U. Straccia, T. Di Noia, E. Di Sciascio, and F.M. Donini. Fuzzy matchmaking in e-marketplaces of peer entities using Datalog. *FSS*, 10(2):251–268, 2009.
14. A. Ragone, T. Di Noia, E. Di Sciascio, F. M. Donini, and Michael Wellman. Computing utility from weighted description logic preference formulas. In *DALT '09*, pp. 158–173, 2009.
15. A. Ragone, T. Di Noia, F. M. Donini, E. Di Sciascio, and Michael Wellman. Weighted description logics preference formulas for multiattribute negotiation. In *SUM '09*, pp. 193–205, 2009.
16. N. Tintarev and J. Masthoff, Designing and Evaluating Explanations for Recommender Systems. In *Recommender Systems Handbook* 479–510, 2011.