Ontology-based Grounding of Spoken Language Understanding

Silvia Quarteroni, Marco Dinarelli, Giuseppe Riccardi

DISI – University of Trento
38050 Povo (Trento), Italy
{silviaq,dinarelli,riccardi}@disi.unitn.it

Abstract—Current Spoken Language Understanding models rely on either hand-written semantic grammars or flat attribute-value sequence labeling. In most cases, no relations between concepts are modeled, and both concepts and relations are domain-specific, making it difficult to expand or port the domain model. In contrast, we expand our previous work on a domain model based on an ontology where concepts follow the predicate-argument semantics and domain-independent classical relations are defined on such concepts. We conduct a thorough study on a spoken dialog corpus collected within a customer care problem-solving domain, and we evaluate the coverage and impact of the ontology for the interpretation, grounding and re-ranking of spoken language understanding interpretations.

I. Introduction

In Spoken Dialog Systems (SDS), the most widespread models of Spoken Language Understanding (SLU) are based on the identification of slots (entities) within one or more frames (frame-slot semantics) defined by the application domain (as in e.g., ATIS [1]). Such a model is limited in two main respects: first, the concept taxonomy is often too domain-specific and must be re-defined when moving towards a new domain; furthermore, there is rarely any account of which relations may occur between concepts and when these are defined, they are generally purpose-built for a specific application.

To address these issues, we advocate the use of an *ontology as a domain model* for a SDS in order to exploit not only knowledge about the properties of individual concepts, but also their relations, expressed in terms of classical semantic relations. An ontology-based representation of the domain concepts has the advantage of (re)using ontologies developed for other domains by the scientific community. In related work, ontologies have been used in the context of SDS to support a variety of objectives: ellipsis and reference resolution in the output of Automatic Speech Recognition [2], representation and clustering of user intentions within a dialog manager [3], or creation of Natural Language Generation rules in a smart home environment [4]. However, the two shortcomings outlined above remain largely true in current SDS technology. Moreover, little work exists to our knowledge about ontology use for SLU: we believe that using an ontology may be very beneficial to validate interpretations by assessing how plausible they are according to the ontology.

Recently [5], we proposed an approach to ontology design and implementation within an SLU module, adding an extra layer of interpretation to the attribute (concept)-value interpretation performed by a baseline SLU system. This is achieved by mapping each concept interpretation to an instance of an ontology concept, thus activating its relations with the other concepts during interpretation. The approach we follow for ontology design is generic and lightweight, making it quickly applicable to different domains. In this paper, we expand our previous work to accommodate a wider range of ontology relations, value-based SLU and a wider application window of ontology relatedness. Moreover, we study the use of ontology relatedness as a re-ranking strategy for SLU interpretations deriving from a probabilistic model. We demonstrate our approach by working with a version of the ontology designed to represent the customer care and technical support center domain as studied within the European project LUNA (ist-luna.eu), and an updated version of its in-domain reference corpus.

This paper is structured as follows. Sec. II describes our approach to ontology modelling and how we interface a concept ontology to the SLU component of a SDS; Sec. III illustrates our experiments to analyze ontology relations in a reference corpus. Sec. IV describes our baseline SLU model and analyzes the relations between ontology relatedness and the results of our baseline SLU module. Sec. V reports on different SLU hypothesis re-ranking experiments based on ontology relatedness. Finally, Sec. VI discusses future work and draws conclusions on our study.

II. Ontology as a Domain Model

While in the past domain modeling for SLU has mainly relied on *ad hoc* concepts with domain-dependent relations, our approach is intended to be generic and portable to other domains. For this reason, we model ontologies as trees rooted in an abstract class `Concept`. Moreover, it appears intuitive to represent the semantics of a domain in terms of the relations between the predicates (actions) and the arguments they take (objects): a notable element of novelty in our model is the fact that it follows the predicate-argument approach, used so far for other types of annotation (e.g., FrameNet-based [6]).

Predicate and argument roles of concepts are represented in our model by two abstract `Concept` subclasses, `PConcept` and `AConcept`. The former represent predicative concepts, i.e., concepts which define an action performed on a number of arguments; for instance, in the LUNA domain,

`HardwareOperation` is performed on an instance of `Hardware`. Classes of concepts that may only be arguments of such predicates are subclasses of `AConcept`; an example of this in the LUNA domain is `Peripheral`.

A. Ontology classes

The concept hierarchy of the ontology created for LUNA contains 32 concept classes, the main ones being illustrated in Table I. In addition, each class has a number of attributes ("slots"): for instance, `Computer` – a `Hardware` subclass – has 2 attributes, `type` (e.g., *PC*) and `brand` (e.g., *DELL*).

TABLE I
TOP CLASSES IN THE LUNA ONTOLOGY TREE AND THEIR FREQUENCY IN THE REFERENCE CORPUS SPLIT USED AS A TEST SET (REF)

PConcept subclass	Example	Freq. (REF)
`GenericAction`	calling	49%
`HardwareOperation`	recording	32%
`SoftwareOperation`	launching	11%
`GenericOperation`	replacing	5%
`NetworkOperation`	connecting	2%
`HardSoftOperation`	restarting	1%
`DocumentOperation`	writing	0%
AConcept subclass	Example	Freq. (REF)
`Hardware`	the printer	19%
`Number`	twelve	36%
`User`	my colleague	15%
`Problem`	printer problem	14%
`Institution`	company X	5%
`Time`	this morning	5%
`Software`	the browser	4%
`GraphicalInterface`	the pointer	2%
`Document`	the folder	1%

The SLU task consists of labeling word sequences as either concept attributes or `null` in case they are irrelevant to the domain; in some cases, SLU also assigns values to concepts, i.e., generalizations of their surface forms into a predefined set of classes. For instance, a correct annotation A for turn $t_1 =$ *"la nostra stampante non stampa più"* (which translates to: "our printer does not print anymore") would be:

$A =$ `Peripheral.type{printer}`{*la nostra stampante*}
`HardwareOperation.negate{}`{*non*}
`HardwareOperation.type{to_print}`{*stampa*}
`null{}`{*più*}.

B. Ontology Relations

Table II summarizes the six different classes of relations modelled in the ontology, which are domain-independent and inspired by classical semantics [7]. Four of these are class-level, while two (negation and attribute-of) are attribute-level. Following such relation description, annotation A for t_1 contains: 1) a *REL SUPER* relation, as there exists a relation between `HardwareOperation` and `Hardware`, the superclass of `Peripheral`, 2) a *NEGATE* relation between the `negate` and `type` attributes of `HardwareOperation`.

C. Ontology Relatedness

The main interest of defining concept relations is evidently to exploit these to infer a notion of concept "relatedness" and

TABLE II
ONTOLOGY RELATIONS APPLICABLE TO A `CONCEPT.ATTRIBUTE` PAIR

Class rel.	Description	Example (a,b)
IS-A (a, b)	class b is class a's super-class (`AConcept` only)	(`Peripheral`, `HardwareComponent`)
SUPER (a, b)	a and b have same super-class (`AConcept` only)	(`NetworkComponent`, `ExternalDevice`)
REL (a, b)	a relation is defined between classes a and b (b is an `AConcept`)	(`ProblemSoftware`, `Software`) (`SoftwareOperation`, `Software`)
REL SUPER (a, b)	a relation is defined between class a and class b's superclass (b is an `AConcept`)	(`HardwareOperation`, `Peripheral`) (`ProblemHardware`, `Computer`)
Attribute rel.	Description	Example (x,y)
NEGATE (x, y)	x negates y (`PConcept` attributes only)	(negate, type) in `HardwareOperation`
ATTR-OF (x, y)	x and y are attributes of the same class (`AConcept` attributes only)	(brand, name) in `Peripheral`

be able to validate an SLU interpretation. Several schemes can be defined to formalize such relatedness; as a simple solution, we defined in [5], a binary pairwise concept relatedness metric. According to the latter, the relatedness between two concepts c_i and c_j, $r(c_i, c_j)$, is defined as equal to a constant R_{MAX} if a relation among those defined in Table II is applicable to (c_i, c_j), and to 0 otherwise. For instance, $r($`ProblemHardware, Software`$) = 0$ as no relation is defined between `ProblemHardware` and `Software`, while $r($`ProblemHardware, Peripheral`$) = R_{MAX}$ as a *REL SUPER* relation holds between `ProblemHardware` and `Peripheral`.

Given a hypothesis H and a concept c_i within it, we can compute the average relatedness of c_i with respect to its neighborhood by averaging the binary relatedness between c_i and the concepts located within a sliding window of size w:

$$r_w^H(c_i) = \frac{1}{|S_w^H(i)|} \sum_{(c_i,c_j) \in S_w^H(i)} r(c_i, c_j), \qquad (1)$$

where $S_w^H(i)$ denotes the set of concept pairs (c_i, c_j) s. t. $(j - i) < w$ in H. If $S_w^H(i) = \emptyset$, then $r_w^H(c_i) = R_{MAX}$. Hence, given annotation A above, $r_2^A($`Peripheral.type`$) = \frac{r(\text{Peripheral.type, HwOp.negate})+r(\text{Peripheral.type, HwOp.type})}{2}$ $= \frac{R_{MAX}+R_{MAX}}{2} = R_{MAX}$, then $r_2^A($`HwOp.negate`$) = \frac{r(\text{HwOp.negate, HwOp.type})}{1} = R_{MAX}$, and $r_2^A($`HwOp.type`$) = R_{MAX}$.

Finally, using $r_w^H(c_i)$, we can define the following utterance-level relatedness metric, representing the combined relatedness among the concepts in H:

$$rel_w(H) = \frac{\sum_{c_i \in H} r_w^H(c_i)}{R_{MAX}}. \qquad (2)$$

In our example, following (2) and choosing $w = 2$, $rel_2(A) = \frac{r_2^A(\text{Peripheral.type})+r_2^A(\text{HwOp.negate})+r_2^A(\text{HwOp.type})}{R_{MAX}} =$

$$\frac{R_{MAX} + R_{MAX} + R_{MAX}}{R_{MAX}} = 1.$$

The occurrences of ontology relations in our reference corpus and the distribution of ontology relatedness scores therein are analyzed in Sec. III.

D. A portable, lightweight approach to domain modeling

An immediate observation that derives from the ontology structure is the following: whenever the need arises to represent a new concept class, we only need to insert the new class in the most suitable position within the ontology tree. For instance, if the current version of the ontology contains an `AConcept` subclass named `Hardware`, with existing subclasses `Computer` and `ExternalDevice`, we can model a new type of hardware such as `SmartPhone` and simply insert it in the ontology tree as a new subclass of `Hardware`. By construction, since the *IS-A*(`SmartPhone,Hardware`) relation holds, `SmartPhone` will inherit the attributes of `Hardware` and all relations which apply to `Hardware` (involving e.g., `ProblemHardware`, `HardwareOperation`) will automatically be extended to the new class. This quick operation allows to extend the ontology to new subdomains without losing consistency.

Moreover, the process from theoretical domain engineering to the extraction of ontology relations from an SLU hypothesis is lightweight in our framework. First, we obtain a direct mapping of the ontology concept tree (coded in Protégé, protege.stanford.edu) into a class hierarchy represented in an object-oriented programming language (Java^{TM} in our case). Here, each program class mirrors an ontology class, directly encoding superclass/subclass relations as well as all relations described as class attributes. For instance, a `HardwareOperation` inherits from the `PConcept` class, and its attributes include the `String` attribute `operationType`, as well as an instance of class `Hardware`. When confronted with a new concept-value interpretation from SLU, the meta-programming feature of reflection is used to create an instance of the corresponding class and exploit its attributes to represent ontology relations.

Consequently, both in terms of design and implementation, the mapping between the ontology representation and the extra layer of interpretation represented in the SLU module is automatic, so virtually this makes the ontology exploitable at no additional cost for an existing SLU module. Another argument in support of the domain-independence of our approach is that, given an existing SLU application with a domain-specific taxonomy (and dataset), the effort required to represent such taxonomy concepts in terms of an ontology of `PConcept` and `AConcept` subclasses is light but nonetheless allows to take advantage of relations (and relatedness metrics). Indeed, this is what we have done to adapt a previously existing taxonomy of concepts (with associated dataset) from the LUNA project for the purpose of our experiments (see Sec. III – V). In addition, the generic relations we defined provide an implicit reasoning about the domain, while the presence of publicly available ontologies, either limited to a particular domain (see e.g. e-tourism.deri.at) or encompassing the whole Semantic Web [8], can completely remove the burden of domain modeling.

III. REFERENCE CORPUS ANALYSIS

Our reference corpus consists of a set of 727 Human-Machine (HM) dialogs (resulting in 4500 turns) containing spontaneous customer requests, recorded by the technical support center of an Italian company. The dialogs follow one of ten possible scenarios inspired by the services provided by the company (e.g., broken printer, email virus) and have been manually transcribed. Transcriptions have been manually annotated using the ontology and split into a training and test set for our statistical SLU model (see Sec. IV). The test set we are using as the reference corpus for SLU results contains 128 dialogs composed of 714 turns, where 1789 non-null concepts have been annotated.

A. Distribution of concepts

Figure 1 illustrates the distribution of the concepts in the reference turns, showing that there are 335 turns carrying at most one concept (typically greetings and short replies), while fewer than 20 turns contain more than 10 concepts. Hence, rel_w is only applicable to the 359 remaining turns.
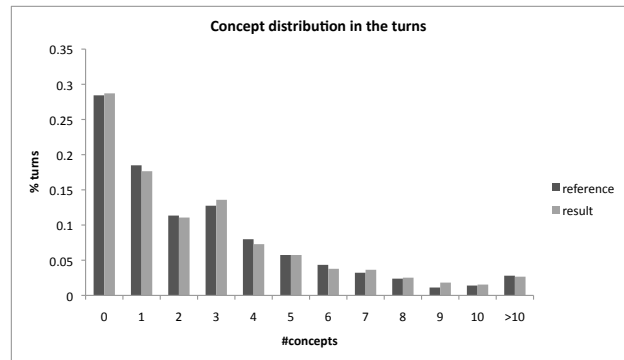

Fig. 1. Distribution of turns per number of concepts (% in the reference corpus and top SLU result

The third column of Table I illustrates the distribution between different types of `AConcept` and `PConcept` in the reference corpus. The most frequent instances of the former refer to either `Number` (as in call codes, tags), subclasses of `Hardware` and `Problem`, and `User` (callers identifying themselves or referring to colleagues). The majority of predicative concepts appearing in the corpus refer to `GenericAction`, referring to actions such as "checking" or "re-trying", while the second most popular `PConcept` is `HardwareOperation`.

B. Distribution of relations and relatedness

The distribution of ontology relations found in the reference corpus is illustrated in Figure 2(a). While hypernymy relations (*IS-A* and *SUPER*) appear rarely, the most frequently occurring relations are those occurring between a `PConcept` and an `AConcept`

or its superclass (such as `HardwareOperation` and `Computer`) or between `AConcept` (such as `ProblemSoftware` and `Software`). The negation relation is also well represented (`HardwareOperation.negate` - `HardwareOperation.operationType`).
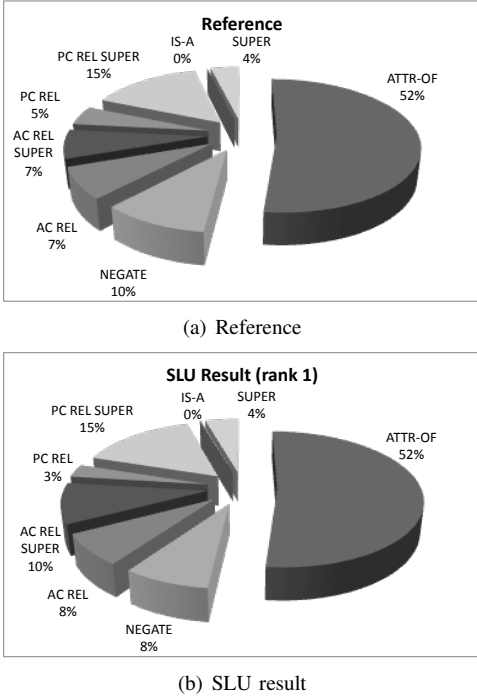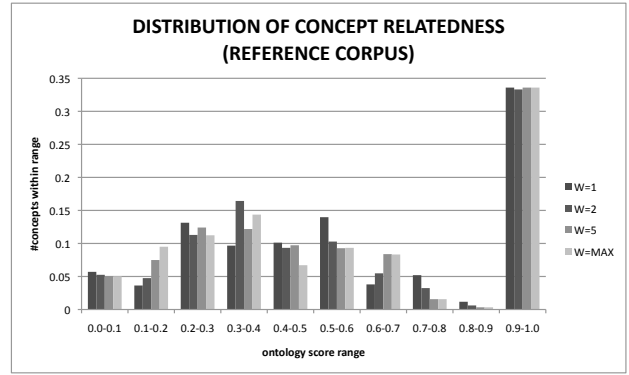

(a) Reference

(b) SLU result

Fig. 2. Distribution of ontology relations in the reference corpus and in the top SLU interpretations (AC = `AConcept`; PC = `PConcept`)

(a) Reference

(b) top SLU result

Fig. 3. Coverage of rel_w in the reference corpus (a) and top SLU result (b) for w varying between 1 and MAX (maximum number of concepts found in the turn). Each column represents the percentage of concepts whose utterance relatedness falls within the underlying range for a given value of w

Figure 3(a) illustrates the coverage of the relatedness metric rel_w in the reference corpus for different window sizes. Coverage is measured in terms of the number of concepts whose rel_w falls in each of 10 consecutive relatedness ranges[1]. It shows that the most significant part of the concepts in the reference annotations tend to occur in sentences that exhibit a high value of rel_w, and this remains true for any value of w. However, it can be noted that the ontology relatedness is not a direct measure of correctness, i.e., not all correct interpretations fall within the $[0.9, 1.0)$ range. This may be explained by considering that, in spontaneous conversation, the concepts mentioned in a sentence need not necessarily be "related". For example, in the reference turn t_2: "*si volevo sapere come mai non è ancora arrivato il tecnico per una chiamata che ho fatto ieri*" ("yes I'd like to know how come the technician hasn't come yet for a call I placed yesterday"), containing the the concepts `User.position` (*technician*) and `Time.relative` (*yesterday*), $rel_{MAX} = 0$.
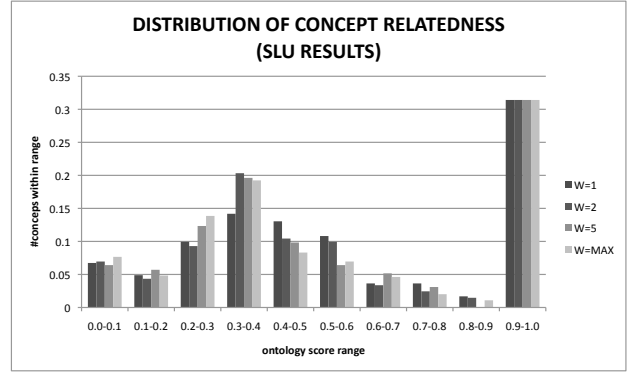
IV. Spoken Language Understanding

We have used our training and test set in order to train our statistical model for SLU, which produces a list of hypotheses

[1]In contrast to our previous experiments [5], we have been focusing on all the relations defined in Table II.

mapping surface words to concepts via Finite State Transducers (FST) encoding a Stochastic Conceptual Language Model (SCLM) [5]. Our model composes three transducers: λ_W, the FST representation of the input sentence, λ_{W2C}, the FST mapping words to concepts, and λ_{SLM}, the SCLM converted into a FST representing the joint probability of word and concept sequences. Composition results in the joint model [5]:

$$\lambda_{SLU} = \lambda_W \circ \lambda_{W2C} \circ \lambda_{SLM}.$$

We use the output of λ_{SLM} to get semantic chunks realizing the concepts used to extract normalized values. Value extraction is performed with a deterministic approach based on hand-crafted rules mapping concept realizations into values.

Having trained λ_{SLU} on the training set described above, we have run it on the test set turns (our reference corpus), obtaining a ranked list of up to ten interpretations (the baseline Concept Error Rate of the top interpretation is 24.8% for attributes and 27.3% for attribute-values, as visible in Table III). The topmost SLU hypotheses contain 1837 concepts altogether, 48 more than in the reference; as visible in Figure 1, the distribution of turns per number of concepts is also very similar to the one in the reference corpus.

A. Initial statistics

Our first analysis on the SLU results was to assess the distribution of ontology relatedness in the top SLU interpretations; the latter, reported in Figure 3(b), shows a similar behavior to the reference corpus (Figure 3(a)). This means that the findings for the reference are grounded by the SLU interpretation and encourage us to study the relation between rel_w and accuracy, as reported in the concept- and value-based statistics sections.

B. Concept-value Precision and Accuracy

We then plotted the concept-value precision of the top SLU result against the ontology relatedness range of the latter. We define concept-value precision as the ratio between the concept-value pairs found in the SLU hypothesis and the total number of concept-value pairs annotated in the reference utterance. Our results, reported in Fig. 4, show that increasing values of rel_w actually correspond to an increased concept precision.
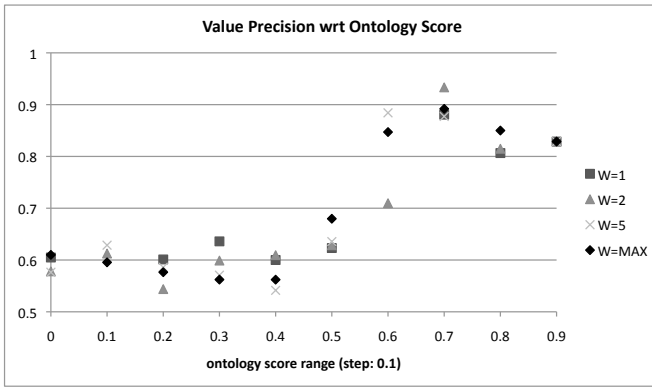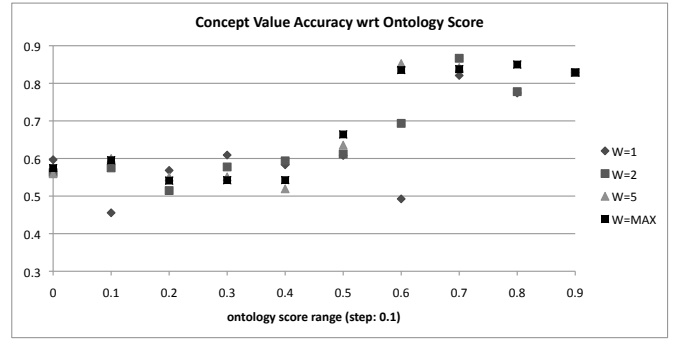

Fig. 5.   Concept value accuracy for different w values

the most similar interpretation to the reference annotation. Figure 5 shows that after an initial decrease, the turn accuracy tends to increase with the ontology score, reaching about 68% in the $[0.8, 1.0]$ range. However, accuracy can further be improved as the oracle result shows.

Fig. 4.   Concept value precision of top SLU hypothesis for different w values

As precision is a "bag-of-concept" measurement, we also measured concept-value accuracy based on the concept-wise Levenshtein distance between the top SLU interpretation and the reference. In particular, the accuracy of a concept-value pair is 1 if the latter occurs in the alignment minimizing the Levenshtein distance matrix, and 0 otherwise. Fig. 5 illustrates the concept-value accuracy for the top SLU hypothesis: although the values are lower than for a bag-of-words metric like precision, the behavior is similar to the previous case, where a growing ontology range is mirrored by a growing accuracy.

Figures 4 and 5 suggest that rel_{MAX} is the choice yielding highest performances; hence, $w = MAX$ is used in the forthcoming experiments.

C. Turn-value Accuracy

We also derived a turn-level value accuracy metric from the Levenshtein edit distance between the SLU interpretation and the reference annotation. Here, the value accuracy of a turn is 1 if the Levenshtein distance between its concept-value pairs is 0, and 0 otherwise. As an upper bound for turn value accuracy, we also plotted the performance of the oracle hypothesis, i.e.,
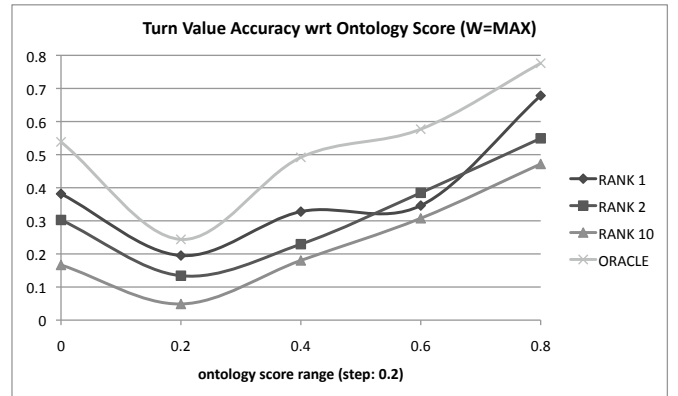

Fig. 6.   Average turn value accuracy for the SLU hypotheses ranked 1, 2 and 10, and for the oracle hypothesis

V. Grounding and Re-ranking SLU hypotheses

A "naïf" approach to re-ranking SLU interpretations based on their ontology relatedness consists of examining the top 10 ranks and returning the first interpretation encountered having a higher relatedness than that of the top rank, or the top rank itself if no such an interpretation is found. Unfortunately, this approach does not prove effective, as illustrated in Figure 7 (cf. "RERANK").

A. Problem statement turns

Further analysis suggests that the turns where ontology relatedness is most meaningful are the problem statement turns, i.e., the ones where callers introduce the problem to be solved. The subsequent turns, which mainly contain answers to questions posed by the customer care interlocutor, are often elliptic (i.e., contain no predicates) or require a dialog context extending beyond the current turn in order to be interpreted. Indeed, when examining the distribution of ontology relatedness in the top SLU hypothesis for the first two dialogue turns

of our corpus, which correspond to the problem statement, we notice a difference with respect to the subsequent turns. While the first two turns have a well-distributed rel_w, the latter tends to have low values in the subsequent turns except for the $[0.9, 1.0)$ range, mainly containing *ATTR-OF* matches (lists of numbers, user name and surname spelled out, …).

Consequent to these observations, we have tried a "special" re-ranking, consisting of applying re-ranking only to the first two turns of each dialog, leaving the others unchanged; however, only a small improvement could be observed compared to naïf re-ranking and none with respect to the baseline (Fig. 7, "SPECIAL").
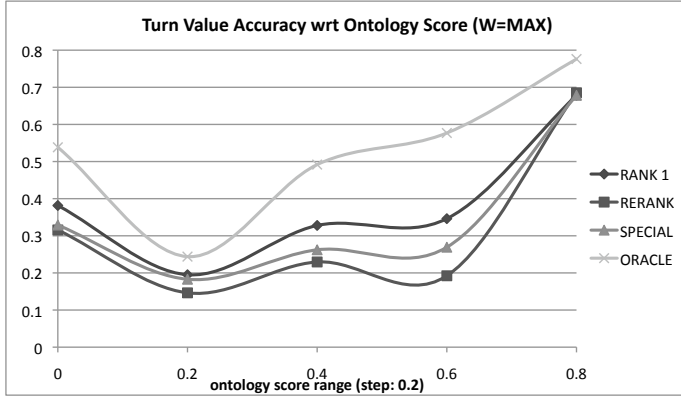

Fig. 7. Average turn value accuracy for the top SLU hypothesis, after "naïf" re-ranking (RERANK) and after re-ranking only on the first 2 turns (SPECIAL)

Table III summarizes our findings so far. The loss in turn accuracy obtained by our best re-ranking model ("SPECIAL") is reflected in a decrease in the precision and recall of the re-ranked system as well as an increase in global CER. Moreover, the Mean Reciprocal Rank of the top 2 resulting hypotheses is also slightly lower than in the baseline case.

TABLE III
PRECISION, RECALL AND CER (TOP HYP), MEAN RECIPROCAL RANK
(TOP 2 HYP): BASELINE (FST) AND AFTER RE-RANKING (RR)

Model		Precision	Recall	CER	MRR@2
FST	concepts	82.0%	84.4%	24.8%	84.9%
	values	76.5%	78.9%	27.3%	81.4%
SPECIAL	concepts	82.0%	84.1%	26.1%	84.5%
	values	76.6%	78.9%	29.4%	81.3%

B. A combined confidence metric

The above-mentioned re-ranking methods do not *jointly* exploit the baseline SLU confidence and the ontology based relatedness. In order to produce a combined confidence metric, we applied multivariate linear regression (MLR) using oracle interpretations to estimate the optimal coefficients combining rel_{MAX} and the FST turn-level confidence, c_{FST}. The resulting combined confidence metric for a hypothesis H is:

$$c_{COM}(H) = \alpha \cdot rel_{MAX}(H) + \beta \cdot c_{FST}(H) + \gamma.$$

However, the results of MLR reveal that the ontology's contribution is negligible: the value for α is -0.001, $\beta = 0.644$ and $\gamma = 0.128$. Hence, it clearly appears that MLR-based re-ranking would at best reproduce the original ranking, based on the $c_{FST}(H)$ parameter.

Based on our current experiments, we can draw the following general conclusion: representing concept relations via an ontology such as the one we have introduced is a valid criterion for grounding the interpretation of a Spoken Language Understanding system. However, this does not directly imply that the notion of relatedness based on such an ontology can be efficiently used to improve the results of any SLU method on any dataset. We believe that this is due both to intrinsic reasons and to our specific case. The intrinsic reason is that there exist correct interpretations showing low ontology relatedness: as mentioned earlier, not all concepts in an utterance need to be related (this issue will be taken into account in future work). In our specific case, we have been dealing with noisy data made evident by the corpus statistics in relation with the baseline results of our SLU system (Table III).

VI. CONCLUSIONS AND FUTURE WORK

We research a novel approach to Spoken Language Understanding that consists of: 1) representing the domain model of a SDS via an ontology of predicate and argument concepts, 2) defining a measure of sentence relatedness based on relations defined in the ontology, and 3) using such a relatedness to ground the consistency of interpretations. Based on a corpus of customer care spoken dialogs, we reinforce our previous findings that highlighted a relation between correct SLU interpretations and ontology relatedness. In addition, we have found that the use of ontology relatedness as a re-ranking criterion for SLU hypotheses does not yield significant improvement (Table III). We are currently studying how to improve such results by the joint use of such a relatedness with the baseline model confidence and we plan to explore data from other domains to further verify our results.

REFERENCES

[1] C. T. Hemphill, J. J. Godfrey, and G. R. Doddington, "The ATIS spoken language systems pilot corpus," in *HLT*, 1990.
[2] D. Milward, G. Amores, T. Becker, N. Blaylock, M. Gabasdil, S. Larsson, O. Lemon, P. Manchon, G. Perez, and J. Schehl, "Integration of ontological knowledge with the ISU approach," TALK Project deliverable 2.1, Tech. Rep., 2005.
[3] S. Young, J. Schatzmann, K. Weilhammer, and H. Ye, "The hidden information state approach to dialog management," in *ICASSP*, 2007.
[4] G. Perez, G. Amores, P. Manchon, and D. Gonzalez, "Generating multilingual grammars from OWL ontologies," *Research in Computing Science*, vol. 18, pp. 3–14, 2006.
[5] S. Quarteroni, G. Riccardi, and M. Dinarelli, "What's in an ontology for spoken language understanding," in *INTERSPEECH*, 2009.
[6] C. Johnson and C. J. Fillmore, "The FrameNet tagset for frame-semantic and syntactic coding of predicate-argument structure," in *NAACL*, 2000.
[7] G. Lakoff, *Women, fire and dangerous things*. Chicago University Press, 1987.
[8] G. Antoniou and F. V. Harmelen, *A Semantic Web primer*. MIT press, 2005.