What's in an Ontology for Spoken Language Understanding

Silvia Quarteroni, Giuseppe Riccardi and Marco Dinarelli

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

Abstract

Current Spoken Language Understanding systems rely either on hand-written semantic grammars or on flat attribute-value sequence labeling. In both approaches, concepts and their relations (when modeled at all) are domain-specific, thus making it difficult to expand or port the domain model.

To address this issue, we introduce: 1) a domain model based on an ontology where concepts are classified into either predicative or argumentative; 2) the modeling of relations between such concept classes in terms of classical relations as defined in lexical semantics. We study and analyze our approach on a corpus of customer care data, where we evaluate the coverage and relevance of the ontology for the interpretation of speech utterances.

Index Terms: Spoken Language Understanding, domain modeling, ontology design, semantic relations

1. Introduction

Spoken language understanding (SLU) addresses the problem of extracting and annotating the meaning structure from spoken utterances in the context of human dialogs [1]. In spoken dialog systems (SDS), the most widespread models of SLU are based on the identification of slots (entities) within one or more frames (frame-slot semantics) defined by the application domain. Such model is limited in several respects: 1) the concept taxonomy is often too domain-specific and must be redefined when moving towards a new domain; 2) 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. We propose a lightweight approach to ontology design and implementation within an SLU module, adding an extra layer of interpretation to the attribute-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. We demonstrate our approach by designing an ontology to represent the customer care and technical support center domain as studied within the European project LUNA (`ist-luna.eu`). However, the approach we follow for ontology design is generic and lightweight, making it applicable to other domains.

This work was funded by the European Commission projects LUNA (contract 33549) and ADAMACH (contract 022593).

1.1. Related work

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 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 in the field of SLU: in contrast, we believe that using an ontology may be very beneficial to validate interpretations by assessing how plausible they are according to the ontology.

In this paper, Sec. 2 describes our approach to ontology design and the resulting ontology for the customer care domain; Sec. 3 describes how the ontology is interfaced to the SLU component of a SDS; Sec. 4 and 5 illustrate our experiments to analyze ontology relations in a reference corpus and their relationship with the outcome of SLU results. Finally, Sec. 6 discusses future work and draws conclusions on our study.

2. Ontology as a Domain Model

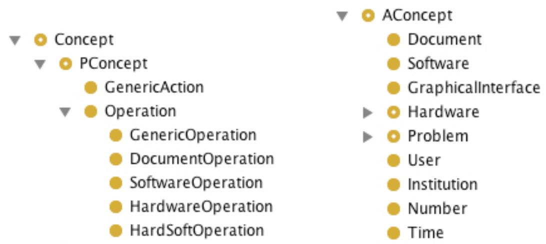
In the past, domain modeling for SLU has mainly relied on *ad hoc* concepts with (optionally, but not always) *ad hoc*, domain-dependent relations. In contrast, our approach to ontology modeling is intended to be generic and portable to other domains. For this reason, we model ontologies as trees rooted 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 which can also be found in other layers of annotation (e.g. FrameNet-based [5]).

Predicative and argumentative 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`.

The concept hierarchy of the LUNA ontology, which has been designed with Protégé[1], contains 32 concept classes, the main ones being illustrated in Fig. 1. In addition, each class has a number of attributes ("slots"): for instance, `Computer` has 2 attributes, `type` (e.g. *laptop, PC*) and `brand` (e.g. *DELL*).

The SLU task consists in labelling word sequences as either concept attributes or `null` in case they are irrelevant to the

[1] Protégé-frames (3.3.1), URL: `protege.stanford.edu`



(a) PConcept (b) AConcept

Figure 1: Structure of the LUNA ontology in Protégé 3.3.1

domain. For instance, the annotation of the following turn:

$$\textit{la nostra stampante non stampa più} \tag{1}$$

which translates to: *our printer does not print anymore*, is:
`Peripheral.type`{*la nostra stampante*}
`HardwareOperation.negate`{*non*}
`HardwareOperation.type`{*stampa*} `null`{*più*}.

2.1. Classical Relations

As mentioned above, our main motivation for using an ontology is to exploit "classical" relations (in the sense of lexical semantics, e.g. in [6]) between concepts and between attributes. Table 1 summarizes the five different classes of relations we consider in the LUNA ontology. For instance, refsentence) contains a REL SUPER relation, as there exists a relation between `HardwareOperation` and `Hardware`, the superclass of `Peripheral`, and a NEGATE relation between the `negate` and `type` attributes of `HardwareOperation`.

Table 1: Classical relations applicable to a concept pair `(a,b)`

Relation	Description	Example
IS-A	`b` is `a`'s superclass (`AConcept` only)	`Peripheral`, `HardwareComponent`
SUPER	`a` and `b` have the same superclass (`AConcept` only)	`NetworkComponent`, `ExternalDevice`
REL	a relation is defined between `a` and `b` (`b` must be an `AConcept`)	`ProblemSoftware`, `Software` `HardwareOperation`, `Computer`
REL SUPER	a relation is defined between `a` and `b`'s superclass (`b` must be an `AConcept`)	`HardwareOperation`, `Peripheral` `ProblemHardware`, `Computer`
NEGATE	negation (`PConcept` only)	`HardwareOperation` `.negate`, `.type`

2.2. Ontology Relatedness

The first step towards assessing the contribution of the ontology to the validation of SLU hypotheses is to associate a binary relatedness measure between nodes in the ontology tree. We define the relatedness between two concepts c_1 and c_2, $r(c_1, c_2)$, as equal to a constant MAX_R if the concepts share a relation among those defined in Table 1, and to 0 otherwise. Different values of MAX_R may be assigned to different classes of relations based on e.g. manual tuning or linear regression from a reference corpus.

Since hypotheses may contain more than two concept interpretations (see Fig. 3), we define the following combined utterance-level relatedness metric. For each concept c_i in a hypothesis hyp, we average the binary relatedness between such concept and the concepts appearing within a given window w:

$$r_w^C(c_i) = \frac{1}{|S_i^w|} \sum_{(c_i,c_j)\in S_i^w} r(c_i, c_j) \tag{2}$$

where S_i^w denotes the set of concept pairs (c_i, c_j) such that $|i - j| < w$. The combined relatedness between the concepts in the hypothesis, $rel_w(hyp)$, is equal to:

$$rel_w(hyp) = \frac{\sum_{c_i\in hyp} r_w^C(c_i)}{MAX\_R}. \tag{3}$$

3. Implementing Ontology Relations

The process from theoretical domain engineering to the extraction of ontology relations from an SLU hypothesis is lightweight in our framework. First, the frame-slot semantics of attribute-value annotation required by SLU is well mirrored by that of Protégé-frames, our chosen ontology editing tool. Moreover, the tool allows to easily represent relations between concepts, which are encoded as attributes having specific types (hence restricting the domain of such relations). For instance, `ProblemHardware` has four attributes:

1. `isRelatedTo`, of type `Hardware`, represents a relation to a subclass of a hardware device;
2. `type`, of type `String`;
3. `ProblemID`, of type `int`;
4. `time`, of type `Time`, represents a relation to the time when the problem occurred.

This paradigm is in turn very similar to the one adopted in object-oriented programming: indeed, we obtain a direct mapping from the Protégé ontology file (.pont) to a Java package, named `lunaOntology`, via a purpose-built parser. This makes any ontology updates extremely easy to transfer to the SLU module.

Each class in the `lunaOntology` package mirrors an ontology class; the advantage is a direct encoding of superclass/subclass relations as well as all relations described as attributes in Protégé. When confronted with a new concept-value interpretation from the SLU, reflection (a meta-programming feature) is used to create an instance of the corresponding class, thus exploiting its properties to represent ontology relations. Figure 2 illustrates the Java class ProblemHardware.java, representing the ontology class `ProblemHardware`.

The first question when using an ontology to either manually or automatically annotate (for model training resp. during SDS deployment) data is how well the former adheres to such data and how well it represents utterance semantics. To address these questions, we analyze our reference corpus in Section 4 and discuss the value of ontology relatedness in our SLU model in Section 5.

```
package lunaOntology;

public class ProblemHardware extends Problem{

Hardware isRelatedTo;
String type;
String ProblemID;
Time time;
  ...
}
```

Figure 2: Java representation of class `ProblemHardware`

4. Corpus Analysis

The LUNA dataset is planned to contain 1000 equally partitioned Human-Human (HH) and Human-Machine (HM) dialogs, recorded by the technical support center of an Italian company. We have currently acquired and manually transcribed 4500 HM turns containing spontaneous customer requests. These follow one of ten possible dialog scenarios inspired by the services provided by the company. Such transcriptions have been manually annotated using the ontology and split into a training and test set for our statistical SLU model (see Sec. 5). The test set contains 128 dialogs composed of 585 turns, where 599 non-null concepts have been annotated. In this section, we analyze the annotation of such test set, which is the reference for SLU results.

Figure 3 illustrates the distribution of the concepts in the reference turns, showing that there are about 210 turns carrying no concepts (typically greetings), and 123 involving only one concept, progressively decreasing until 12. Hence, relatedness based on the ontology is only applicable to the remaining turns.

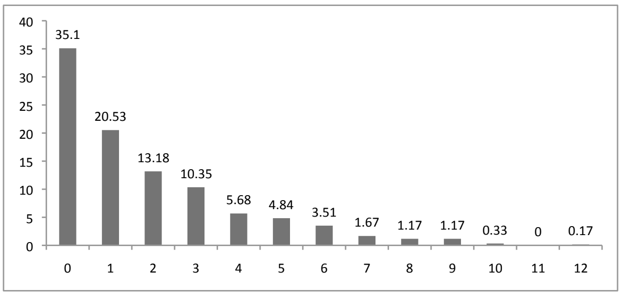


Figure 3: Distribution of concepts per turn in the reference (%)

Figures 4 and 5 illustrate the distribution between different types of `AConcept` and `PConcept` in the reference corpus. The most frequent instances of the former refer to either `User` (callers identifying themselves or referring to colleagues), or subclasses of `Hardware` and `Problem` (indeed, these often cooccur in the reference). Moreover, the majority of predicative concepts appearing in the corpus refer to hardware operations (`HardwareOperation`), while the second most popular `PConcept` is `GenericAction`, referring to actions such as “checking” or “re-trying”.

The distribution of ontology relations found in the reference corpus is illustrated in Figure 6. There is a total of 1542 instances of relations between concepts, in addition to 2870 pairs of concepts sharing the same class (e.g. `Time.day` and `Time.month`). While hypernymy relations (IS-A and SUPER) appear rarely, the most frequently occurring relations are those occurring between a `PConcept` and an

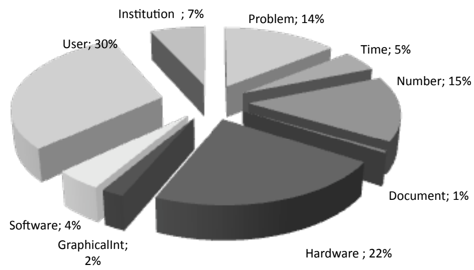


Figure 4: `AConcept` distribution in the reference corpus

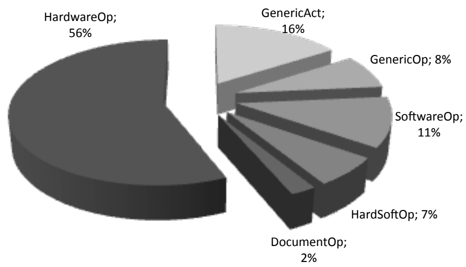


Figure 5: `PConcept` distribution in the reference corpus

`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`).

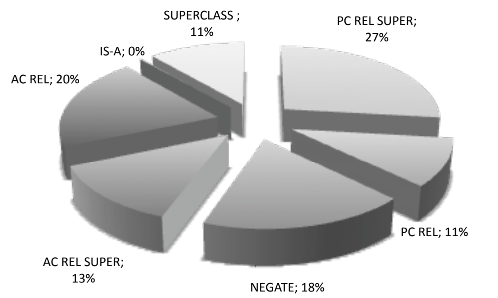


Figure 6: Distribution of ontology relations in the reference corpus (AC = `AConcept`; PC = `PConcept`)

Figure 7 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[2]. As it can be noted, 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 understood by considering that in spontaneous speech, the concepts mentioned in a sentence need not necessarily be “related”, even in a loose sense. Indeed, in most cases the relatedness lies in (or above) the $[0.6, 07)$ interval.

[2]In our experiments, we have been focusing on relations, excluding `AConcept` hypernymy (i.e. IS-A and SUPER), which account for less than 12% of relations, from the relatedness metric.

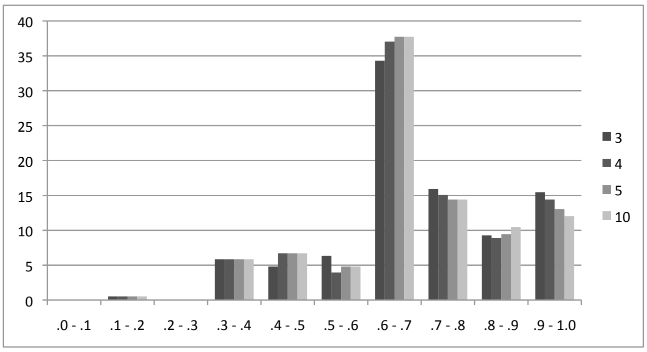


Figure 7: Coverage of the utterance relatedness rel_w in the reference corpus for w varying between 3 and 10. Each column represents the percentage of concepts whose utterance relatedness falls within the underlying range for a given value of w

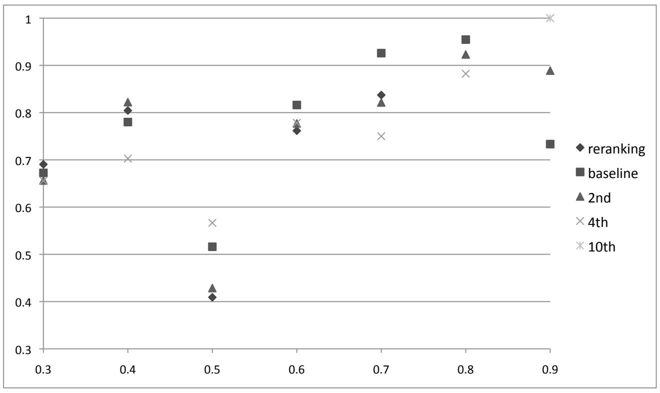


Figure 8: Probability of correct interpretation with respect to rel_5 range for the first ranked SLU hypothesis ("baseline"), the next ranked hypotheses, and after reranking

5. Spoken Language Understanding

We have used our training and test set in order to train our statistical model for Spoken Language Understanding, which produces a list of hypotheses mapping surface words to concepts via a Stochastic Conceptual Language Model (SCLM). Our model is a slight modification of the one described in [7]: the main difference is that we train the language model and we then convert it into a stochastic Finite State Transducer (FST). The λ_{SLU} model therefore combines three transducers:

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

where λ_W is the transducer representation of the input sentence, λ_{W2C} is the transducer mapping words to concepts and λ_{SLM} is the SCLM converted into an FST. The latter represents the joint probability of word and concept sequences: $P(W, C) = \prod_{i=1}^{k} P(w_i, c_i|h_i)$, where $W = w_1..w_k$, $C = c_1..c_k$ and $h_i = w_{i-1}c_{i-1}..w_1c_1$.

Having trained the SLU model on the training set described above, we have run it on the test set turn, obtaining a ranked list of up to ten interpretations (the baseline Concept Error Rate of the top interpretation is 27.1%). We then measure the frequency of correct concept interpretations with respect to the ontology relatedness interval in the top ten SLU results. We define such correctness to be the number of matches (in terms of both concept and surface[3]) between the reference utterance annotation and the SLU hypothesis and the total number of concepts annotated in the reference utterance.

Figure 8 shows that after an initial erratic behavior (partly due to data sparseness), the probability of correct interpretation tends to increase with the ontology relatedness. Such behavior is visible in all the ranks of the SLU hypotheses and remains true after reranking (see below).

In addition, we experimented with a first attempt to perform ontology-based reranking. This consists in iterating over the top 10 ranks and returning: (a) the first interpretation encountered having a higher relatedness than that of the top rank (provided that such relatedness falls within $[0.6, 0.9)$, the relatedness interval in which it is more likely to observe correct interpretations according to our corpus analysis), or (b) the top rank itself if no such interpretation is found.

[3] surface match is relaxed to accept the case where the annotation surface is included in the hypothesis surface or *vice versa*

The results of Figure 8 do not imply a direct use of the ontology relatedness as a confidence metric to influence SLU re-ranking (cf "reranking"). Indeed, we are currently studying the conversion of relatedness into a suitable confidence metric to combine with the one deriving from our statistical model.

6. Conclusions and Future Work

We have introduced a novel approach to Spoken Language Understanding that consists in representing the domain model of a SDS via an ontology of predicative and argumentative concepts and leveraging the classical semantic relations defined therein to produce a measure of sentence relatedness and validate the consistency of interpretations. We have defined an ontology following such model to suit a customer care domain and experimented with in-domain speech data. Based on these, we have highlighted the existence of a relation between correct SLU interpretations and ontology relatedness. We are now studying a conversion of the ontology relatedness into a suitable confidence metric to be combined with the baseline confidence.

7. References

[1] R. De Mori, F. Bechet, D. Hakkani-Tur, M. McTear, G. Riccardi, and G. Tur., "Spoken language understanding: A survey," *IEEE Signal Processing magazine*, vol. 25, no. 3, pp. 50–58, 2008.

[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 ISUs 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 *Proceedings of ICASSP'07*, 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] M. Dinarelli, S. Quarteroni, S. Tonelli, A. Moschitti, and G. Riccardi, "Annotating spoken dialogs: from speech segments to dialog acts and frame semantics," in *Proceedings of SRSL'09*, 2009.

[6] G. Lakoff, *Women, fire and dangerous things*. Chicago University Press, 1987.

[7] C. Raymond and G. Riccardi, "Generative and discriminative algorithms for spoken language understanding," in *Proceedings of Interspeech2007*, Antwerp,Belgium, 2007.