

# ONTOLOGY GROUNDING

*Aleks Jakulin, Dunja Mladenić*  
Department of Knowledge Technologies  
Jozef Stefan Institute  
Jamova 39, SI-1000 Ljubljana, Slovenia  
Tel: +386 1 4773900; fax: +386 1 477 31 31  
e-mail: aleks.jakulin@ijs.si

## ABSTRACT

**An ontology is a structured semantic model, composed of concepts, relations and instances. Data is a more primitive but concrete assembly of instances described by their attributes. An example of topic ontology is Open Directory of Web documents used in Google organizing documents into topics and subtopics, such as “Science”, “Arts”, etc. However, many ontologies are originally manually developed without making an explicit connection to the data. We hereby introduce the concept of ontology grounding, where the concepts and relations from the ontology become associated with the data. This enables us both to explain concepts in more concrete terms and to recognize them in the data.**

## 1 INTRODUCTION

In this paper we introduce the process of ontology grounding, whereby a concept of the ontology becomes recognizable in the underlying data (such as database records and text). This process is especially suitable for problem domains where extensive data is available, and where it would be time consuming to manually convert unstructured data into structured metadata.

Although the key technology that underlies ontology grounding is machine learning, ontology grounding should be seen as a process which also involves knowledge engineering and information extraction. For example, an ontology might be excessively abstract, and as such ungroundable in data; it would be very hard to recognize all instances of the concept “event” or “object” from text: such concepts can only be built upon lower levels of abstraction. On the other hand, the data itself must be conceptualized in some way, through information extraction. For example, the text can be represented using the standard bag-of-words representation, or we could parse it and identify different grammatical entities. Therefore, ontology grounding is the process of associating abstract concepts to the concrete data. The task of knowledge engineering is to provide groundable concepts, and the task of information extraction is to extract supporting concepts. Machine learning merely connects these efforts from both directions.

In the remainder of the paper we will elaborate upon the definition of ontology grounding, and relate to similar problems that have been discussed primarily in philosophy. A more concrete approach follows in section 3, with the description of machine learning methodology. Finally, we will report our experience on a practical application of ontology grounding, and take a look into the future.

## 2 GROUNDABLE AND CONCEPTUALIZABLE

The notion of grounding is an old question, which has been elaborated in the context of artificial intelligence by S. Harnad [1], but the concepts have been around in philosophy already in the times of C. S. Peirce [3]. The fundamental notion is that the higher-level abstract concepts are grounded in lower-level concrete concepts, which, in turn, are grounded in perceptions.

Association goes in two ways: there are well-defined concepts that are not grounded in data (“unicorn”), and there are patterns in the data that are not yet captured by a dedicated concept (“the eating of pizza while watching a sunset”). The concept of a unicorn is *groundable*, however: we would recognize a unicorn had we seen one, we could even generate hypothetical data.

Similarly, the eating of pizza while watching the sunset is *conceptualizable* and could be denoted by a new hypothetical concept “sunspizaing”. Of course, only a sufficient amount of data is unambiguously conceptualizable: having been given a single picture of “a female with a child eating the pizza during the sunset on a street with a red sports car passing by”, and told to learn the concept of this single picture, we could identify a number of possible concepts: “a person”, “sunset”, “street”, “car”, “junk food”. We would thus be unsure about what to make of it. Therefore, a complex concept requires plenty of data and/or additional background knowledge.

In summary, a *grounded* concept is one that is both conceptualizable (there is data exemplifying the concept) and groundable (the concept can be recognized from the data).

The background knowledge can even be a selection of the words in this case: the interpretation is less ambiguous once we pinpoint a few key words. For example, once we indicate that “a person”, “sunset” and “pizza” are key while “street” and “car” are not, the concept becomes defined. Similarly, should we lack the concept of “pizza”, we would have to define it beforehand, as otherwise the concept of “sunspizaing” would not be grounded. Thus, background knowledge is composed of both the underlying concepts and of attention windows.

One should distinguish ontology grounding from unique identification. A particular concept can be identified uniquely for example using the uniform resource identifier or locator (URI/URL) [2]. However, this does not assure that the grounding of that concept is also unique. For example, while we may agree on the term “dog” and thus have a unique identification, we might disagree about the grounding – whether a domesticated wolf is a dog or not.

### 3 CLASSIFICATION

The previous section has discussed the problem of grounding in a rather abstract fashion. We will now present the usual problem of classification in machine learning and examine how it relates to the problem of grounding a topic ontology. In contrast to the usual problem of classification in machine learning, we will not question the learning algorithm, but the representation of the data and especially the concept structure.

The problem of classification has been addressed in machine learning and in statistics for a very long time. Recently, these technologies have been applied to text. A number of methods have been proposed, but for the past few years the support vector machines (SVM) have been considered to be the state-of-the-art. In addition to this, we used one-to-one handling for multi-category classification and probabilistic SVM. Through extensive benchmarks, this configuration outperformed other methods.

Even if the learning methodology is state-of-the-art, the results of learning might not be satisfactory. Of course one could passively blame it all on learning, but a more active approach would examine the causes of learning failure, and attempt to remedy them. For this idea to work, we need to attempt to present the failure in as much detail as possible to the knowledge engineer. We will now present methods for the identification and analysis of the grounding failure.

#### 3.1 Description of Data

As a part of the SEKT project (European 6FP integrated project), the next generation knowledge management technologies will be evaluated on several case studies including a legal domain case study. The core problem of this case study is to enhance question answering and

information retrieval tasks through additional semantic information. A novice judge will type a query and the system should return answers that match the query. In a preliminary study [8], a number of likely queries have been gathered through interviews with the judges. The queries have been manually organized into a topic ontology roughly as follows:

|     |                               | #   | error |
|-----|-------------------------------|-----|-------|
| 1.  | On Duty                       | 79  | 0.23  |
| 2.  | Family                        |     |       |
|     | a. Family violence            | 62  | 0.29  |
|     | b. Minors                     | 19  | 0.34  |
|     | c. Divorces                   | 18  | 0.32  |
|     | d. Other                      | 4   | 0.36  |
| 3.  | Foreigners                    | 26  | 0.35  |
| 4.  | Property                      | 10  | 0.33  |
| 5.  | Sentences                     | 2   | 0.36  |
|     | a. Execution                  | 40  | 0.27  |
|     | b. Noncompliance              | 4   | 0.36  |
|     | c. Form                       | 1   | /     |
|     | d. Notification               | 1   | /     |
| 6.  | Process                       | 197 | 0.00  |
|     | a. Trial                      | 61  | 0.25  |
|     | b. Competency                 | 45  | 0.27  |
|     | c. Criteria                   | 4   | 0.34  |
| 7.  | Office                        |     |       |
|     | a. Organization               | 40  | 0.31  |
|     | b. Officers                   | 17  | 0.35  |
|     | c. Financial                  | 20  | 0.32  |
|     | d. Infrastructure             | 19  | 0.33  |
|     | e. Security                   | 2   | 0.37  |
|     | f. Information Technology     | 13  | 0.35  |
|     | g. Assessment                 | 9   | 0.35  |
|     | h. Police                     | 11  | 0.35  |
|     | i. Lawyers                    | 2   | 0.35  |
|     | j. Incompatibilities          | 14  | 0.34  |
|     | k. Ministry of Justice (CGPJ) | 2   | 0.34  |
|     | l. Other                      | 2   | 0.37  |
| 8.  | Trade and Business            | 28  | 0.31  |
| 9.  | Traffic Accidents             | 11  | 0.34  |
| 10. | Penal                         | 5   | 0.32  |
|     | a. Drugs                      | 3   | 0.34  |
|     | b. Theft                      | 2   | 0.37  |

Next to each concept, we list the number of questions in the database that correspond to it. The error in grounding these concepts will be explained at the end of Section 3.2. It can be seen that this ontology identifies the key types of problems faced by a judge. On the other hand, each query is formulated as a question, for example:

*How to proceed in a case where a woman files a complaint for ill treatment, but specifically requests that the restraining order is not to be issued?*

There are currently 773 similar questions in the database. To ground this ontology, we need to be able to classify a particular question in the form of text into the appropriate semantic concept(s) in the above ontology. For example, the above question indicates an instance of the process/jurisdiction concept. This defines our grounding problem. We represent the question using the standard bag-of-words text representation with TFIDF weighting [7], preprocessing the Spanish text using a lemmatizer [6].

With a grounded ontology, we would be able to identify the topic area of the question. This can be useful for narrowing down the set of candidate matches. For example, if the question is clearly a process/jurisdiction one, we can provide answers only from that topic area. In that sense, we have extracted relevant semantic information, which helps perform semantic matching. The linking of ontologies with words or terms has been anticipated earlier [8], but the application of classifiers proposed here automatically finds the relevant terms and adjusts their weights. Instead of providing the linking directly, one can simply mark the questions up with concepts, and leave the rest to the classifier.

### 3.2 Identifying Grounding Errors

Since each question in the data is classified into exactly one category, the categories are mutually exclusive and we have simply a multiclass classification problem. But simply attempting to classify a question into the correct class would result in many errors. It is preferable to rank the classes by their suitability. If we employ probabilistic classification, we obtain a probability for each concept. That probability indicates how likely it is that a particular concept is the best descriptor for the question. Usually, multiple concepts obtain a nonzero probability. For the above example of a question, the dominant predictions of classes with the corresponding probabilities are as follows:

*Process* (0.311), *Process/Trial* (0.123), *On Duty* (0.094)

The main class of the question is predicted to be *Process*, with a possible refinement into a *Trial* question, and an additional class of *On Duty*.

But is this classification correct? How can we evaluate the correctness of such probabilistic classification? Although a single question might conceivably belong to multiple classes, the training data only lists the single one that best describes it. Partly this is because the purpose of labeling was only to organize the questions, not to provide their semantic markup. For that reason, maximizing the probability of the “correct” class might not be the best approach to identifying mistakes. For instance, in an ambiguous case assigning  $p_c=0.1$  to the correct class might not be a bad choice if the second-highest class probability is 0.05: the correct class has twice the probability of any other. In another case, we might assign the correct class  $p_c=0.3$  (which is more than 0.1 earlier), but the wrong class would be assigned  $p_w=0.7$  (which is far more than 0.05 earlier). The naïve approach of probability maximization would deem the second case a greater error than the first one.

For that reason, the loss or error function will be defined as the difference between the highest class probability and the probability assigned to the correct class. Performing leave-one-out validation to prevent overfitting, we can thus

identify the problematic classifications. The ontology is well-grounded when few or no classification errors occur. The results are shown in the topic ontology in the previous page. It can be seen that the rare concepts are not identified well: the Process concept seems to overwhelm other concepts.

### 3.3 Explaining Grounding Errors

Once the problematic classifications have been identified, actions must be taken to remedy them. By focusing only on the grounding errors, we already save a considerable amount of work. Only the non-trivial and problematic instances deserve our attention.

The actions can be improvements to the extraction of instances and features, or the provision of background knowledge. But it may also turn out that the ontology is inappropriate, preventing sensible classification. Or, the reference classification may not be correct. To choose the best action, it is helpful to provide insight into the causes of the errors.

Any grounded ontology contains a classification model able to recognize the concept in the data. Although the support vector machine itself can be expressed explicitly [4], it is often complex. Instead, we can explain the classification of a particular problematic instance, providing arguments in favor and against the classification. Following the methodology of [4], we can express a SVM based on a dot product kernel in the form of a linear model:

$$t = b + \beta_1 x_1 + \beta_2 x_2 + \dots$$

Each of the algebraic terms  $\beta_i x_i$  belongs to a specific word or an underlying concept,  $b$  is the bias and  $t$  is the distance from the separating hyperplane. To explain this classification, we can pick the most distinct terms (most positive, most negative), and present them in a list. For example, in the above case, the true classification (*Domestic Violence*) can be reasoned for and against as follows:

#### Keywords in favor of family/domestic violence

0.28 (1) alejamiento [restraining]  
 0.18 (1) pedir [file]  
 0.14 (1) denunciar [complain]  
 0.01 (1) trato [treatment]

#### Keywords against family/domestic violence

0.19 (1) caso [case]  
 0.17 (1) mujer [woman]  
 0.13 (1) pero [but]  
 0.07 (1) ninguno [no]

The original Spanish terms are accompanied by the corresponding term in English. Bracket contains the number of appearances of the word, and the number in front indicates the weight of the term corresponding to the word. In this case, the classifier determined that the

question is *not* an instance of the Domestic Violence concept. There are several causes to that:

- “Restraining order”, “file a complaint” and “ill treatment” are not considered to be concepts of their own, to be identified in a sentence. In fact, these concepts are well-known in legal terminology (feature generation, feature construction).
- “But” and “no” are not relevant arguments in this case (feature selection).
- The question is a good example for the Domestic Violence concept (instance selection).

Therefore, such explanation helps focus the actions needed to assure a greater quality of concept grounding. We can see that feature generation is nothing else than the grounding of underlying concepts.

#### 4 ONTOLOGY ENGINEERING WORKFLOW

The inclusion of grounding affects how the ontology is designed. Our proposal is shown in Fig. 1. Instead of using merely human expert feedback to adjust the ontology, we include grounding in the process. Specifically, grounding provides feedback both towards the designers of the ontology schema, and toward the designers of the information extraction modules.

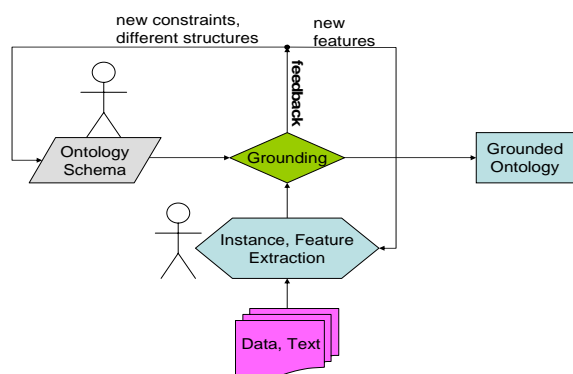


Figure 1: The process of grounding provides feedback for both ontology design and information extraction.

#### 5 CONCLUSION

We have shown that the ontology can be linked with the data through the process of ontology grounding. This is an application of the common probabilistic classification techniques. When the learning algorithm cannot reliably recognize the appearance of concepts in text, we must question both the ontology and the data. On one hand, we may remedy the problem by making an ontology more concrete and less abstract. On the other hand, we can improve the features that are extracted from the data as to capture a higher level of abstraction, or we can adjust the way the instances are extracted from text.

In the present work we focused on topic ontologies, which are trivial in terms of instance extraction: each question or document is an individual instance. In a more complex situation, the extraction of instances is something to be considered. We could have used glossaries such as WordNet in order to expand the number of the features extracted from text: each individual sense would correspond to a feature extracted from text.

The grounding of concepts is a rather simple case of ontology grounding. In the future, it will be necessary to ground relations and slots. The techniques for these tasks are already under development in the machine learning community. Because infrequent concepts may be discriminated against frequent ones, the machine learning method should properly deal with imbalanced data sets.

#### Acknowledgement

This work was supported by the Slovenian Research Agency and the IST Programme of the European Community under SEKT Semantically Enabled Knowledge Technologies (IST-1-506826-IP), and PASCAL Network of Excellence (IST-2002-506778). This publication only reflects the authors' views.

#### References

- [1] Harnad, S. (1990) *The Symbol Grounding Problem*. Physica D 42: 335-346.
- [2] Berners-Lee, T., Fielding, R., Mastiner L. *Uniform Resource Identifier (URI): Generic Syntax*, RFC 3986, January 2005.
- [3] Peirce, C. S. (1868) *On a New List of Categories*. Proceedings of the American Academy of Arts and Sciences 7 (1868), presented 14 May 1867.
- [4] Jakulin, A., Možina, M., Demšar, J., Bratko, I., Zupan, B., (2005) *Nomograms for visualizing support vector machines*. In Proceedings of the 11th ACM SIGKDD international conference KDD-05, 108-117.
- [5] Gabrilovich, E., Markovitch, S. (2005) *Feature Generation for Text Categorization Using World Knowledge*. In Proceedings of the IJCAI-05, Edinburgh, Scotland.
- [6] Carmona, J., Cervell, S. Márquez, L. Martí, M.A., padró, L. Placer, R., Rodríguez, H., Taulé, M. & Turmo, J. (1998) *An Environment for Morphosyntactic Processing of Unrestricted Spanish Text*. In Proceedings of LREC'98, 915-922, Granada.
- [7] Sparck Jones, K. (1972) *A statistical interpretation of term specificity and its application in retrieval*. Journal of Documentation, Vol. 28, pp. 11-21.
- [8] Benjamins, V.R., Contreras, J., Casanovas, P., Ayuso, M., Becue, M., Lemus, L., Urios, C. (2003) *Ontologies of Professional Legal Knowledge as the Basis for Intelligent IT Support for Judges*, Workshop on Legal Ontologies and Web-based Legal Information Management, held at ICAIL 2003, Edinburgh, June 2003.