Property-based Semantic Similarity: What Counts?

Silvia Likavec and Federica Cena

Università di Torino, Dipartimento di Informatica, Torino, Italy
{likavec,cena}@di.unito.it

Abstract. Similarity, one of six Gestalt principles, is one of the most intuitive ways to perceive the world and categorise the objects surrounding us. The notion of similarity plays an important role in many areas, and it is important to simulate human perception of similarity in order to obtain satisfying results in various applications. We draw our inspiration from Tversky's work on similarity and define property-based similarity for ontological concepts taking into account their common and distinctive features and their values. We also discuss some possible ways to improve the property-based similarity.

Keywords: Similarity, properties, ontology

1 Introduction

It is inherent to human nature to try to categorize objects surrounding us, finding patterns and forms they have in common. One of the most intuitive ways to relate two objects is through their *similarity*. Similarity is one of the six Gestalt principles which guide the human perception of the world, the remaining ones being: Proximity, Closure, Good Continuation, Common Fate, and Good Form.

According to Merriam Webster "similarity" is a quality that makes one person or thing like another and "similar" means having characteristics in common. There are many ways in which objects can be perceived as similar, such as having similar color, shape, size, texture etc. But if we move away from just visual stimuli, we can apply the same principles to define the *semantic similarity* of two objects. This leads to a similarity based on features these two objects have in common, and consequently, the lack of distinctive features characterising each object.

The concept of semantic similarity can be encountered in various fields, from Natural Language Processing (NLP) and Information Retrieval to Semantic Web. In this work we deal with the semantic similarity of concepts in domain ontologies (Gruber, 1993, Guarino and Poli, 1995), where concepts are distinguished by the properties associated to them. The usage of ontologies to represent various domains accounts for both similarities and differences among domain objects as well as generic objects and very specific ones.

Our inspiration comes from Tversky's work on Features of Similarity (Tversky, 1977) and we try to apply his ideas to similarity among ontological objects. More precisely, two objects are similar if they both are defined having the same properties with the same values. In addition to this simple notion of similarity, we explore how this similarity can be improved by considering relevance for properties or relevance for values or hierarchical relationships among values, Throughout this work we use domain of recipes to provide examples and explain our approach and reflections.

The rest of the paper is organised as follows. In Section 2, we provide a brief background on ontologies for knowledge representation and on the treatment of properties in OWL. We give the details of how to calculate the property-based similarity for instances in the domain ontology in Section 3 and then we look into some possible ways to improve the property-based similarity in the ontology in Section 4. We summarise the most relevant related work which regards the semantic similarity in Section 5. Finally, we conclude in Section 6.

2 OWL ontologies and knowledge representation

In various fields, from e-commerce and e-learning to cultural heritage, medicine, digital libraries etc., it is possible to describe the concepts of the domain by using the properties of these concepts and their respective values. The ones that immediately come to mind are ontologies (Antoniou and van Harmelen, 2008, Allemang and Hendler, 2008) and linked open data (Bizer et al., 2009), where properties are prominent elements of the domain and contribute to the description of domain concepts.

In this work we deal with ontologies, powerful and expressive formalisms which make it possible to explicitly specify domain elements and their properties, as well as relationships which exist among domain elements. Also, rigorous reasoning mechanisms are associated with ontologies. One standard formalism for representing ontologies is OWL.[1]

Throughout this work we would use domain of recipes to provide examples and explain our approach and reflections.

2.1 Properties in OWL

In ontologies expressed in OWL *properties* are used to describe domain elements and express their features. There are two kinds of properties in OWL:
(i) *object properties* describing relations among individuals and
(ii) *data type properties* providing relations among individuals and data type values.
Object properties and datatype properties are defined as instances of the built-in OWL classes owl:ObjectProperty and owl:DatatypeProperty, respectively. Both are subclasses of the RDF class rdf:Property. Here, we only consider object properties, and leave the treatment of data type properties (such as literal values) for future analysis, since it is more complex.

The *property axiom* is used to define the characteristics of a property. Usually, it defines its *domain* and *range*. rdfs:domain links a property to a class description, whereas rdfs:range links a property to either a class description or a data range. For example:

```
<owl:ObjectProperty rdf:ID="has_ingredient">
  <rdfs:domain rdf:resource="#Recipet"/>
```

[1] http://www.w3.org/TR/owl-ref

```
  <rdfs:range rdf:resource="#Food"/>
</owl:ObjectProperty>
```

defines a property has_ingredient which connects the elements of Recipe class to the elements of Food class.

Equivalent properties are defined with owl:equivalentProperty.

Properties can be explicitly defined for the classes and can be used to define classes with property restrictions. Our approach to similarity is best illustrated when considering instances in the ontology, hence we will provide here a brief description of properties for instances.

2.2 Instances and their properties

An instance in the ontology are characterised by its class membership, individual identity and property values. An instance inherits its properties from the classes it is an instance of and it has a specific value associated to each property. For example:

```
<Recipe rdf:ID="Herbed_Asparagus">
  <has_ingredient rdf:resource="#Asparagus"/>
  <has_ingredient rdf:resource="#Parmesan"/>
  <has_ingredient rdf:resource="#Herbs"/>
  <has_origin rdf:resource="#Italy"/>
  <suitable_for_diet rdf:resource="#"Vegetarian"/>
</Recipe>
```

defines a recipe Herbed_Asparagus which has ingredients: asparagus, parmesan and herbs, originates from Italy and is suitable for vegetarians.

3 Property-based similarity

First of all, let us have a look at an example which should clarify the basics of our approach. We consider the domain of recipes where properties such as has_ingredient, has_origin, suitable_for_diet are defined. These properties have one or more values assigned to them. Intuitively, the similarity among recipes depends on the property-value pairs they have in common. Consider for example the following recipes: Asparagus_Parmigiana and Herbed_Asparagus_With_Parmesan_Cheese. They both have ingredients: Asparagus, Butter, Parmesan, Pepper among others and are both suitable for vegetarian diet. On the other hand, Indian_Style_Chicken has only Butter in common with any of them and is not suitable for vegetarians. So the asparagus dishes are definitely more similar among themselves than any of them with the chicken dish.

Hence, in order to determine similarity among two objects, we want to consider both, their common features and distinctive features for each of them. To this aim we use Tversky's feature-based model of similarity (Tversky, 1977):

$$\textsc{sim}_T(O_1, O_2) = \frac{\alpha(\psi(O_1) \cap \psi(O_2))}{\beta(\psi(O_1) \setminus \psi(O_2)) + \gamma(\psi(O_2) \setminus \psi(O_1)) + \alpha(\psi(O_1) \cap \psi(O_2))}. \quad (1)$$

where $\psi(O)$ is the function describing all the relevant features of the object O, and $\alpha, \beta, \gamma \in \mathbb{R}$ are constants which permit different treatment of the various components. For $\alpha = 1$ common features of the two objects have maximal importance and for $\beta = \gamma$ non-directional similarity measure is obtained. In our approach we have $\alpha = \beta = \gamma = 1$.

We will be using the following notation:

- *common features of* O_1 *and* O_2*:* $\textsc{cf}(O_1, O_2) = \psi(O_1) \cap \psi(O_2)$,
- *distinctive features of* O_1: $\textsc{df}(O_1) = \psi(O_1) \setminus \psi(O_2)$ and
- *distinctive features of* O_2: $\textsc{df}(O_2) = \psi(O_2) \setminus \psi(O_1)$.

Using this notation and setting $\alpha = \beta = \gamma = 1$ the formula (1) becomes:

$$\textsc{sim}_T(O_1, O_2) = \frac{\textsc{cf}(O_1, O_2)}{\textsc{df}(O_1) + \textsc{df}(O_2) + \textsc{cf}(O_1, O_2)}. \tag{2}$$

Since each of the domain objects has a number of property-value pairs describing it, for each property p we will have to calculate how much it is responsible for common features among these objects, as well as for distinctive features of each of them. We denote these values by $\textsc{cf}_p$, $\textsc{df}^1_p$ and $\textsc{df}^2_p$. We consider equal the properties defined with owl:EquivalentProperty.

3.1 Similarity among instances

In this work we present our approach only for instances of classes, although it can be extended to classes defined with their properties and to classes defined as property restrictions (see Cena et al. (2012)). The essence of property-based similarity calculation lies in simple comparison of the property-value pairs for each instance. Let us assume that the property p has h' different values in O_1 and h'' different values in O_2, and k is the number of times O_1 and O_2 have the same value for p, then

$$\textsc{cf}_p = \frac{k^2}{h'h''}, \textsc{df}^1_p = \frac{h'-k}{h'} \text{ and } \textsc{df}^2_p = \frac{h''-k}{h''}.$$

Let us assume that the objects O_1 and O_2 have properties $p_1, \ldots, p_n$ in common. We can repeat the above process for each property $p_i, i = 1, \ldots, n$.

Now, there are two possible ways to calculate similarity between O_1 and O_2.

First, we can obtain all common and distinctive features of O_1 and O_2:

$$\textsc{cf}(O_1, O_2) = \Sigma^n_{i=1} \textsc{cf}_{p_i} \qquad \textsc{df}(O_1) = \Sigma^n_{i=1} \textsc{df}^1_{p_i} \qquad \textsc{df}(O_2) = \Sigma^n_{i=1} \textsc{df}^2_{p_i}$$

where n is the number of properties O_1 and O_2 have in common. The similarity between two instances O_1 and O_2 is then calculated using the formula (2):

$$\textsc{sim}(O_1, O_2) = \frac{\textsc{cf}(O_1, O_2)}{\textsc{df}(O_1) + \textsc{df}(O_2) + \textsc{cf}(O_1, O_2)}.$$

This method for property-based similarity of objects in the ontology was first introduced in (Cena et al., 2012) classes defined with property restrictions but only for value restrictions. It was further developed to include cardinality restrictions and applied to categorization of shapes in (Likavec, 2013).

Second, we can calculate partial similarities w.r.t. each property $p_i, i = 1, \ldots, n$:

$$\textsc{sim}_{p_i} = \frac{\textsc{cf}_{p_i}}{\textsc{df}^1_{p_i} + \textsc{df}^2_{p_i} + \textsc{cf}_{p_i}}$$

and then use these similarities to calculate the total similarity between O_1 and O_2 as:

$$\textsc{sim}(O_1, O_2) = \Sigma^n_{i=1}\textsc{sim}_{p_i}.$$

4 Improving property-based similarity

The above presented base case property-based similarity provides high rates of similarity among objects which can be used in many applications. We still did not perform the thorough evaluation but we evaluated it in the field of user interest propagation and obtained very satisfying results (Cena et al., 2012). But, while performing the second evaluation in this field, we became aware that in certain domains, this property-based similarity of domain objects can be improved w.r.t. various aspects. We will discuss here some of them.

4.1 Relevance of properties

When defining the concepts of a domain, not all the properties play an equal role. Hence, it is possible to introduce the *relevance of properties* and assign different importance to different properties in the domain. Actually, the relevance of a property can be considered as the capacity of the property to determine the similarity between two entities. For example, in the recipe domain, the property `has_ingredient` is far more important than `has_author` and the two recipes with the same ingredients would be considered more similar than the two recipes with the same author. So, the property `has_ingredient` would have a higher relevance factor than prophas_author.

There are various approaches to calculation of property relevance in a domain. It can be declared *a priori* and although effective, this solution may not be very feasible for a huge domain. Also, it is possible to introduce an automatic method to determine the relevance of properties. One possibility is to compute the similarity of concepts and then to calculate the relevance factor for each property as the square of the average similarity between concepts with the same value for that property.

4.2 Property or underlying hierarchy?

First of all, some aspects of the domain can be seen as properties, as well as underlying hierarchy. So the question is, which way of modelling of the domain would provide better similarity with human judgement. For example, in the recipes domain, the concepts corresponding to dish type can be easily organised into a hierarchy and we can have all the instances be instances of certain `Dish_Type` classes. On the other hand, we can simply have a property `dish_type` and have all the recipes be instances of `Recipe`.

4.3 Relevance of values

One of the problems with the approach in which all the values for properties are treated equally is that they might not contribute to the overall similarity with the same degree, since some values might be more important in a certain context than the other. For example, if we consider recipes and the `has_ingredient` the values `beef` or `asparagus` would be more important than `salt` or `pepper`. Hence, we come to the point where we might need to introduce relevance for values, along the lines for relevance for properties. These would have to be proposed by domain experts or calculated by an algorithm designed for this purpose.

4.4 Hierarchy of values

Another possible improvement of property based similarity is to take into account the underlying hierarchy which might exist among the concepts used as values for properties. For example, if we consider recipes and the property `has_ingredient`, one recipe can have ingredient `Fusilli` and the other one `Spaghetti`. Although these two concepts are not equal, they could be considered equal or equal to a certain degree (e.g. 80% equal), since they are both types of pasta, and are descendants of `Pasta` concept. So it might be possible to consider "almost equal" direct descendants of a certain concept and even less equal second degree descendants of a certain concept.

5 Related work

There are various approaches to calculating similarity among concepts, depending on the data structure used to represent the domain and on the amount and type of data available about the concepts of the domain. The principal approaches to similarity calculation are the following: (i) information content-based methods, (ii) distance-based methods and (iii) feature-based methods. Various hybrid methods combine some of the above methods.

In his seminal paper, Resnik (1999) proposes to calculate the semantic similarity of concepts by calculating the information content in an is-a taxonomy of the closest class subsuming both compared concepts. This similarity measure is given by the negative logarithm of the probability of occurrence of the class in a text corpus. Another important information-theoretic definition of similarity is introduced by Lin (1998) where the similarity among concepts is calculated taking into account the shared information for the two concepts and the amount of information needed to fully describe them.

The origins of the distance-based approach go back to Rada et al. (1989) where the ontology graph structure is used to calculate the *distance* between nodes (i.e., the number of edges or the number of nodes between the two nodes) as a measure of their similarity. Leacock and Chodorow (1998) use the normalised path length in WordNet (Fellbaum, 1998) between all the senses of the concepts being compared. The semantic similarity is computed as a negative logarithm of the ratio between the number of nodes in the path which connects the given concepts and the maximum depth of the taxonomy. Wu and Palmer (1994) take into account the depths of the given words in the taxonomy and the depth of their common subsumer in their similarity measure.

Pirrò and Euzenat (2010) introduced a FaITH semantic similarity measure which uses Tversky's feature-based model and calculates the saliency of the features using a new information content approach based on the ontology structure. This new framework permits to calculate semantic similarity, as well as semantic relatedness and can be used to rewrite the existing similarity measures so that they can also compute semantic relatedness.

Smyth (2007) calculates the similarity by taking into account individual features of concepts and by assigning to each feature its own similarity function and the weight which helps distinguish the importance of individual features..

A semantic similarity measure for OWL objects introduced by Hau et al. (2005) is defined as a ratio between the shared and total information content of the two objects. The information content is calculated from the objects' description sets containing all the statements describing the given objects and is based on the number of new RDF statements that can be generated by applying a certain set of inference rules to the predicate.

The similarity measure introduced by Zadeh and Reformat (2013) is similar to ours in the sense that it uses Tversky's feature-based model for calculating similarity and then calculates object's common and distinctive features by observing all the relations the objects have in the given ontology.

In the realm of "Conceptual Spaces" proposed by Gärdenfors (2004) the concepts can be seen as convex regions in a conceptual space, whereas instances correspond to points. The conceptual spaces are constructed using primitive quality dimensions which represent various qualities of objects (e.g., color, shape, size). These dimensions of conceptual spaces provide the means for determining similarity between concepts and instances which can be defined as the inverse of their distance in the space.

Recently, Conceptual Spaces have been integrated with ontological formalisms to form hybrid knowledge bases by Lieto et al. (2015). Since the points are represented as vectors of the point coordinates (representing various object dimensions), their mutual similarity is calculated as cosine similarity.

6 Conclusions and future work

In this work we present an approach to calculate similarity based on properties defined in an ontology, as well as insights on which other factors can be included to improve this similarity in different contexts. We limited ourselves to presenting the approach only for the instances in the ontology, although the approach can be applied to classes and classes defined as property restrictions as well. In addition, this approach can be applied to linked open data Bizer et al. (2009) or any other structure where the objects are described by means of their properties. For example, it would be interesting to apply our measure of similarity to ConceptNet Speer and Havasi (2013), where an edge which connects two nodes can be seen as a property and a target concept as its value.

In the case presented here, the prerequisite is the ontology with explicitly defined properties for classes, rather than only a simple taxonomy of concepts. We only dealt with object type properties in this work, since data type properties, such as literals, require more complex analysis.

One of the limitations of the present approach, known for Tversky's notion of similarity, is that in the case of concepts with few properties defined for them, is is possible that some concepts would be equally similar to the concepts which in reality have different degrees of similarity with them. This problem can be overcome by enlarging the knowledge base with as many properties as possible for each concept. Also, by assigning relevance to certain properties, the more important features would be taken into account.

The evaluation of the approach on different datasets is being carried out and would be published elsewhere.

Bibliography

Allemang, D. and Hendler, J. (2008). *Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL*. Morgan Kaufmann Publishers.

Antoniou, G. and van Harmelen, F. (2008). *A Semantic Web Primer, second edition*. The MIT Press.

Bizer, C., Heath, T., and Berners-Lee, T. (2009). Linked Data - The Story So Far. *International Journal on Semantic Web and Information Systems*, 5(3):1–22.

Cena, F., Likavec, S., and Osborne, F. (2012). Property-based interest propagation in ontology-based user model. In *20th Conference on User Modeling, Adaptation, and Personalization, UMAP 2012*, volume 7379 of *LNCS*, pages 38–50. Springer.

Fellbaum, C., editor (1998). *WordNet: An Electronic Lexical Database*. MIT Press.

Gärdenfors, P. (2004). *Conceptual spaces: The geometry of thought*. MIT Press.

Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition Journal*, 5(2):199–220.

Guarino, N. and Poli, R. (1995). Editorial: The role of formal ontology in the information technology. *International Journal of Human-Computer Studies*, 43(5-6):623–624.

Hau, J., Lee, W., and Darlington, J. (2005). A semantic similarity measure for semantic web services. In *Web Service Semantics Workshop at WWW (2005*.

Leacock, C. and Chodorow, M. (1998). *Combining local context and WordNet similarity for word sense identification*, pages 305–332. In C. Fellbaum (Ed.), MIT Press.

Lieto, A., Minieri, A., Piana, A., and Radicioni, D. P. (2015). A knowledge-based system for prototypical reasoning. *Connection Science*, 27(2):137–152.

Likavec, S. (2013). Shapes as property restrictions and property-based similarity. In Kutz, O., Bhatt, M., Borgo, S., and Santos, P., editors, *2nd Interdisciplinary Workshop The Shape of Things*, volume 1007 of *CEUR Workshop Proceedings*, pages 95–105. CEUR-WS.org.

Lin, D. (1998). An information-theoretic definition of similarity. In *15th International Conference on Machine Learning ICML '98*, pages 296–304. Morgan Kaufmann Publishers Inc.

Pirrò, G. and Euzenat, J. (2010). A feature and information theoretic framework for semantic similarity and relatedness. In *9th International Semantic Web Conference, ISWC '10*, volume 6496 of *LNCS*, pages 615–630. Springer.

Rada, R., Mili, H., Bicknell, E., and Blettner, M. (1989). Development and application of a metric on semantic nets. *IEEE Trans. on Systems Management and Cybernetics*, 19(1):17–30.

Resnik, P. (1999). Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *Journal of Artificial Intelligence Research*, 11:95–130.

Smyth, B. (2007). Case-based recommendation. In *The Adaptive Web, Methods and Strategies of Web Personalization*, volume 4321 of *LNCS*, pages 342–376. Springer.

Speer, R. and Havasi, C. (2013). The peoples web meets nlp. In Gurevych, I. and Kim, J., editors, *ConceptNet 5: A large semantic network for relational knowledge*, pages 161–176. Springer Berlin Heidelberg.

Tversky, A. (1977). Features of similarity. *Psychological Review*, 84(4):327–352.

Wu, Z. and Palmer, M. (1994). Verbs semantics and lexical selection. In *32nd Annual Meeting on Association for Computational Linguistics*, pages 133–138. Association for Computational Linguistics.

Zadeh, P. D. H. and Reformat, M. (2013). Assessment of semantic similarity of concepts defined in ontology. *Information Sciences*, 250:21–39.