Feature-based vs. Property-based KR: An Empirical Perspective

Massimo Poesio and Abdulrahman Almuhareb
University of Essex
Department of Computer Science and Centre for Cognitive Science
United Kingdom

Abstract. Two types of knowledge representations are most commonly used in Natural Language Processing (NLP) to represent properties of objects. In semantic networks representations, description logics, and typed feature structures, properties are uniformly represented as feature[1] / value pairs: e.g., *an ancient signet-ring* would be represented as $\exists\, x$ **signet-ring**$(x) \wedge$ **age**(x)=*ancient*. In first-order and higher-order representations, the property of being 'ancient' is typically not decomposed: the representation of 'an ancient signet ring' would be $\exists\, x$ **signet-ring**$(x) \wedge$ **ancient**(x) (in FOL representations), or possibly –viewing **ancient** as a predicate modifier–as $\exists\, x$ **ancient**(**signet-ring**)(x), especially for non-intersective adjectives. We have been exploring the advantages and disadvantages of the two types of representation using a combination of large-scale data analysis and machine learning. We found, first of all, that not all 'properties' of objects we find can easily be decomposed into feature / value pairs, which supports the 'property' view. However, we also found that clustering using features works better than clustering using all properties; and that clustering using both features and (certain) properties works best of all.

Introduction

Two types of knowledge representations are most commonly used in Natural Language Processing (NLP) to represent properties of (nominal) concepts. In semantic networks representations, description logics, and typed feature structures, properties are represented as feature / value pairs: e.g., *an ancient signet-ring* would be represented as $\exists\, x$ **signet-ring**$(x) \wedge$ **age**(x)=*ancient*. In first-order and higher-order representations, the property of being 'ancient' is not necessarily decomposed this way: the signet ring would be represented as an $\exists\, x$ **signet-ring**$(x) \wedge$ **ancient**(x) (in FOL representations), or possibly –viewing **ancient** as a predicate modifier–as **ancient**(**signet-ring**)(x).

The arguments typically used in the debate have to do with (linguistic) expressivenes, computational complexity, and intended applications. Description logic-based representations [1] have very good complexity properties (classification is typically polynomial given certain restrictions to the representation language). On the other hand, from a linguistic perspective it has often been noted that these representations cannot easily be used to represent non-intersective adjectives (*former*, *fake*, etc.); nor is it clear how one could reduce to the feature / value format properties such as *flamboyant* (e.g., *that flamboyant signet ring*).

In this paper we discuss preliminary results from our work on concept learning from corpora. The goal of this work is to automatically identify properties and features of objects

---

[1] We use here the terms 'feature' and 'attribute' interchangeably.

(i.e., to find that rings can be ancient, or that they have an age), and to use these properties to cluster concepts into classes (e.g., to find that rings and bracelets are 'more similar' than rings and cats, or bracelets and cats). We believe that work of this kind may provide a new perspective from which to analyze the differences between the two view of styles of concept representations.

The structure of the paper is as follows. We begin by discuss how we extract and analyze the modifiers of nominal concepts. This analysis suggests that not all 'properties' of such concepts can easily be decomposed into feature / value pairs, which supports the view of properties typically taken in formal semantics. We then discuss the methods we are used to identify the features of these concepts. Next, we show the results obtained when using these different types of concept descriptions for clustering. We found that clustering only using features works better than clustering using all modifiers; and that clustering using both features and (certain) properties works best of all. Finally, we discuss a number of questions concerning the notion of 'feature' or 'attribute' raised by this work.

1   Background: Concept Learning from Corpora

In recent years, there has been a great deal of interest in NLP in using corpora to learn ontologies. This work is in part motivated by practical needs–particularly in domains such as biology or medicine in which new types of objects are continuously discovered–in part motivated by psychological research on semantic priming [2, 3].

The simpler type of concept learning methods, originally based on techniques developed in Information Retrieval but these days most famously exemplified by Latent Semantic Analysis [2], are based on the assumption that 'the meaning of a word is specified by the company it keeps'. In this research, the representation of a word w (typically, a vector) indicates which other words w co-occurs with this, possibly with an indication of how strong the correlation between w and each of these other words.

A second approach, pioneered by [4] and pursued more recently by [5, 6], among others, aims at build concept descriptions based on features and relations, like those typically seen in KR work. The crucial simplifying assumption made in this work is that these concept descriptions are built using syntactic information only, such as that produced by parsers like RASP. So, for example, the description of the concept 'car' obtained in this work might be as shown in Table 1.

	modifier Fast	modifier Red	modifier Big	object of Drive	subject of Hit	....
Car	50	34	13	79	7	....

Table 1: A description of the concept **car** in terms of modifiers and syntactic relations

2   Concept Properties: A Preliminary Analysis

In a preliminary experiment, we tried to find the most common properties of concepts by looking for their most common adjectival and nominal modifiers, adopting a simplified version of the methods used in work such as [4, 5, 6]. We chose to study the concepts used by [3] in their experiments aiming at finding concept clusters that would match the results of semantic priming experiments. This list includes 37 concepts: animals such as **cow** and **mouse**, body parts such as **nose**, and countries such as **Brazil**. We initially use the British National Corpus (BNC) to collect modifiers, but we then switched to the Web, accessed via the Google API [7, 8, 9]. The size of the Web enormously reduces the data sparsity problem,

also allowing us to use fairly crude patterns, although we took care to ensure that all of our patterns satisfy Hearst's criteria for 'good patterns' [10]: they are (i) frequent, (ii) precise, and (iii) easy to recognize.

The Google API allows wildcard patterns of the form `"w1 * w2"`, where * matches a single word. To search for properties of the concepts we were studying, we used the pattern:

```
"[a|an|the] * C [is|was]"
```

(double quotes included), where `C` denotes a concept, and the wildcard denotes a single word. (An example match is *an inexpensive car*). Note that, because we are not using any linguistic analysis, not even a POS tagger, we have placed restrictions on the used patterns (i.e., using *is* and *was*)) to make sure that the `C` which stands for a concept is separated from the remaining context. This search of course finds all sort of modifications (more than 1500 for each concept); in order to find the strongest correlations, we run t-tests [11, 6].

The 50 properties of **nose** with the highest–all significant–associations (for significance above .01, values of t above 2.326 are sufficient –cfr. [11], p. 609) are shown in Figure 1.

electronic	16.84	snub	4.66
runny	16.77	rounded	4.61
bloody	10.71	canine	4.48
long	10.48	vehicle	4.45
human	9.52	flat	4.42
broken	9.29	spotted	4.40
stuffy	8.90	bullet	4.33
cat	8.82	running	4.29
spindle	8.04	pink	4.24
wet	7.63	fuselage	4.23
roman	7.52	fractured	4.20
brown	6.77	pointed	4.14
red	6.54	pug	4.14
large	5.95	blunt	4.12
artificial	5.85	dogs	4.09
aircraft	5.81	aquiline	4.01
black	5.73	blake	4.00
preferred	5.47	wax	3.98
false	5.31	bleeding	3.96
hooked	5.31	prominent	3.67
crooked	5.11	insulator	3.50
split	5.09	sensitive	3.48
external	5.05	cold	3.47
blocked	5.02	clown	3.45
good	4.77	entire	3.42

Figure 1: The properties of *nose* with the strongest associations

An analysis of this list of modifiers is very instructive. First of all, not all modifiers express properties of noses: for example, many express concepts which have noses as parts (*aircraft*, *fuselage*, *bullet*). Others express idioms or collocations (e.g., *brown*, as in *brown nose*). Of the modifiers that do express properties of **nose**, some can be viewed as expressing values of features of the concept **nose**, such as shape (*crooked*, *pug*, *flat*), color (*pink*, *red*) or size (*large*). But many of the most common modifiers of *nose* express properties that do not lend themselves easily to a feature / value formulation. Some of these modifiers specify the type of nose used in the particular context: these include, as well as well-known exceptions such as *false*, modifiers like *electronic*, *human*, and *artificial*.[2] And a few of these modifiers express

---

[2]It could be argued that these modifiers simply specify the sense of **nose** under consideration, but it's not clear how this type of interpretation could be derived compositionally in a feature / value framework, except by having separate entries for each sense. (It is well-known that properties like *false* do not lend themselves to an intersective interpretation.) In a framework allowing for predicate modifiers, the derivation could be specified in terms of modification and meaning postulates.

properties of noses that could only be viewed as 'values' of 'features' by making violence to the notion of feature to varying degrees. The best example of this is *prominent*–a common description of certain noses, but not one that is easy to describe as the value of a particular feature, except perhaps a boolean one. Of course, the designer of an ontology could always introduce an *ad hoc* feature–say, 'salience'–but it's not clear to us what generalization this kind of feature could express. These properties that are easily to recognize as typical of noses, but are more difficult to view as expressing values of features, become more frequent as modifiers with decreasing values of t are considered: examples include *adorable*, *mature*, *trained*, etc., as well as more general properties such as *good*. *Bloody*, *broken* and *runny* express typical states of noses, but not ones that are easy to couch in terms of properties of noses, except again by introducing pretty artificial states (e.g., a 'wholeness state' with values 'whole' / 'broken'). An explicit representation of states as objects would be a solution, except of course that, e.g., 'broken' has a different meaning for the case of nose that, say, for vases.

The T test does a reasonably good job at discriminating intrinsic properties of noses from accidental properties. If we look at modifiers with lower t values (i.e., expressing weaker associations), we find, in addition to less common properties of noses (and modifiers that would not be considered with richer syntactic information) a number of 'accidental' properties that, while less clearly describable as properties of the concept **nose**, can clearly be attributed to instances of the concept: examples include *attack*, *folded*, *mobile*, and *slick*. It is clear that to develop a really robust understander it would be necessary to find ways of attaching these other properties, as well.

A preliminary conclusion that can be drawn from this initial analysis, then, is that while some of the properties of concepts like *noses* (in fact, some of the properties with strongest associations) can be naturally viewed as values of features such as shape, size or color, for many others it becomes progressively more difficult to do so. In other words, that a feature / value representation does capture useful generalizations, but it would be too restrictive as the only form of representing properties. A more general question is how can we decide what should count as a 'feature'.

3 Looking for Concept Features

We concluded in the previous section that while perhaps not all modifiers can be viewed as values of features, quite a few are. How can we find such features? There are two basic techniques: clustering feature values (which, however, means finding them–as we just saw, not all modifiers of concepts can be seen as values), or using again the techniques described above, trying to identify syntactic constructions suggesting that certain nouns describe attributes of given concepts. We followed the second approach in this work.

As in the case of the preliminary search for properties above, we used the Web as a corpus to look for concept features, without syntactically analyzing the text. The pattern we use is based on the linguistic test proposed by [12] as a necessary condition for attributes, and was already used by us elsewhere to extract functional nouns [13]:

```
"the * of the C [is|was]"
```

where, as above, `C` denotes a concept, and the wildcard denotes a single word. (An example match is *the size of the nose is*). The 50 relational and functional nouns with the strongest associations with **nose**, and that therefore we consider best potential features, found using this pattern are shown in Figure 2. The t-test only indicated a significant association for 22 'features'; we write 0 as value of t for the other nouns, which are however ranked by frequency:

skin	15.52	complexity	0.00
side	9.56	job	0.00
inside	9.13	light	0.00
base	8.96	top	0.00
shape	8.56	radius	0.00
bottom	8.00	thickness	0.00
end	7.59	narrator	0.00
root	6.89	skeleton	0.00
colour	6.23	bone	0.00
structure	5.25	anatomy	0.00
width	5.23	smell	0.00
function	4.86	breadth	0.00
interior	4.81	sense	0.00
purpose	4.78	volume	0.00
appearance	4.34	object	0.00
floor	4.00	angle	0.00
point	3.54	bill	0.00
center	3.15	support	0.00
length	3.09	blocking	0.00
framework	3.06	effectiveness	0.00
line	2.85	altitude	0.00
size	2.64	purity	0.00
color	0.00	pressure	0.00
height	0.00	installation	0.00
back	0.00		

Figure 2: The features of *nose* with the strongest associations

As can be seen by Figure 2, the pattern we used extracts, in addition to what we may think of as features of noses such as *shape*, *colour*, *structure*, *width*, and *function*, also information about other relations, particularly about their parts (e.g., *skin*, *side*, *inside*, *base*, *bottom*) [14, 13]. For simplicity, we will consider parts as features in this work, as also proposed in work such as [15]. The t-test works relatively well at filtering out accidental relations, such as *narrator*, but it also excludes a number of potentially useful attributes, such as *anatomy* or *breadth*. These observations suggest the need to improve our understanding of the notion of feature, which may allow us to decide whether to consider parts as features or not, as well as to design a more sophisticated test of 'feature-hood', which would allow us to treat as features potential features with weaker associations, while at the same time keeping out irrelevant associations. We return on the issue of what counts as a good feature below.

4 Clustering

We now found two fairly simple ways of extracting characterizations of concepts: one based on all properties, and one based only on features. A simple way to compare these two ways of characterizing concepts is to use them for clustering, i.e., to find which concepts are most similar. In this section we summarize some of the work in [16].

After testing a number of clustering algorithms, including CobWeb [17] (as implemented in the Weka library) and SUBDUE, we settled on the CLUTO 2.1 clustering tool [18], which implements a non-hierarchical agglomerative clustering algorithm. The input to the clustering tool is a frequency table with concepts as rows and values attributes, or the combination as columns. Each cell in the table contains the frequency of co-occurrence between the concept and corresponding value or attribute. Just as done before, before presenting the table to the clustering tool the raw frequencies are transformed into weighted values using the TTest

method as described in [11]; [6] found TTest to be the best weighting method. We call the `vcluster` command of CLUTO with the following parameters: similarity function [Extended Jaccard Coefficient], clustering criterion [Group Average], Clustering Method [Graph Partitioning], and No. of Clusters [Three]. We use the Jaccard similarity function, which has been show to give best results in similar tasks as suggested also by [6]. The results are shown in Table 2. The accuracy of clustering is measured here by computing the percentage of concepts which are put in the cluster containing the majority of concepts of their 'correct' class, divided by all concepts. Additional evaluation methods are discussed in [16], including the precision / recall measure proposed by [19].

Concept	Vector Size				
Characterization	500	1522	3044	4753	4969
Values Only	64.86 %	94.59 %	-	-	94.59%
Attributes Only	97.30 %	97.30%	-	97.30%	-
Attributes and Values	-	-	100.00%	-	-

Table 2: Clustering accuracy for features, values, and their combination

Table 2 shows clustering accuracy when using values, features, and both features and values, with different sizes of the vector describing each concept. The first interesting result is that clustering with attributes leads to better results, and with less information. The table shows that when using vectors of size 500, clustering using attributes is more accurate (97.30%) than clustering using values (64.86%). When the vector size is increased to 1500, the accuracy with features remains the same, whereas the accuracy using only values improves to 94.59%, but it remains lower than the accuracy using only features. In other words, features have more discrimination power than values; with only a 500 vector size attributes produce a very accurate result and values are a way behind. A second result is that increasing the size of the concept description more and more does not improve the accuracy. Finally, and most interestingly, we get the best accuracy (100%) when we combine data about both values and features, forming a data vector of size 3044 (1522 attributes +1522 values).

These results suggest that the notion of 'feature' or 'attribute' does have a usefulness as a way of clustering together separate properties; but also that combining features and values gives us the best classification.

5   What is a feature?

The discussion above raises again the issue of what is a 'feature' / 'attribute' [15]. The features used in KR practice often feel very artificial: an attempt to adjust the knowledge we want to represent to the format we are using, rather than the other way around. This is even the case with with linguistically and psychologically motivated attempts at developing a semantic network, such as WordNet [20].

WordNet makes a distinction between two types of adjectives: RELATIONAL (= derived from nouns), like *musical*, *woody*, and DESCRIPTIVE [21]. Although no association between noun synsets and adjectival meanings is present in the WordNet database (indeed, providing such a link is one of the motivations of our work), the characterization of descriptive adjectives in WordNet is centered around attributes: e.g., "descriptive adjectives typically ascribe to a noun a value of an attribute" ([21], p. 48). The meaning of descriptive adjectives is characterized by BIPOLAR STRUCTURES based on the notion of antonymy [22]: adjectives like *swift*, *prompt*, *alacritous*, *quick*, and *rapid* form a HALF CLUSTER associated by semantic similarity to a FOCAL ADJECTIVE like *fast*, in turn related by a relation of antonymy to the focal adjective–*slow* of a second half cluster, containing adjectives like *sluggish*, *laggard* or *tardy*. Each such pair of half clusters defines an attribute (in this case, **speed**).

Such strong assumptions about the centrality of attributes result in the introduction of a number of attributes whose sole purpose seem to serve as the name for clusters of adjectives not associated with any natural attribute: thus we find, besides natural attributes such as **sex**, **age**, **temperature**, attributes such as **offensiveness** (label for the cluster of *offensive* and *inoffensive*), **auspiciousness / propitiousness** (as a label for the cluster *auspicious-propitious*, *inauspicious-unpropitious*), or **quality** (label for the cluster of *positive-good*, *negative-bad*).

Is there a way to get a good angle on the notion of 'feature'? This question, and the related question of what is a 'role', has long been debated in KR [12]. Given the results we have reported, this is not only a philosophical question; the ability to properly characterize which properties should be viewed in terms of features and values, and which as atomic predicates, may lead to improved results at concept classification.

Ultimately there may not be a completely clear-cut way of characterizing the notion of feature, but attempts in this direction have been made both by philosophers and by linguists. Some of this work only specifies necessary conditions: e.g., the test proposed by [12] that influenced our choice of a pattern. Perhaps the best known attempt at 'cleaning up' the ontological notions used in KR is the work by Guarino, inspired by philosophical criteria. In [15], a distinction between concepts, roles, slots and 'attributes' is proposed; the notion of attribute introduced there is the closest to our notion of feature. In the final version of the paper we discuss this definition and examine its feasibility for our purposes.

In future work, we plan to investigate the possibility of developing ontologically sounder ways of identifying attributes, and to evaluate such improved characterizations using the methods discussed in this paper.

6 Discussions and Conclusion

We analyzed the properties attributed to objects, finding that many of them cannot be naturally interpreted as 'values' of 'features'. On the other hand, quite a few can, and identifying the features of concepts appears to add to the characterization of a concept, as shown by the fact that concept descriptions based on attributes lead to better results at clustering than concept descriptions including only values. On the other hand, the best descriptions of all for these purposes include both features and additional properties. In conclusion, we would like to argue that corpus- and machine learning techniques such as those used here are a promising way to shed light on traditional issues in KR and in the philosophy of concepts.

References

[1] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider, editors. *The Description Logic Handbook*. Cambridge, 2003.

[2] T. K. Landauer, P. W. Foltz, and D. Laham. Introduction to latent semantic analysis. *Discourse Processes*, 25:259–284, 1998.

[3] K. Lund, C. Burgess, and R. A. Atchley. Semantic and associative priming in high-dimensional semantic space. In *Proc. of the 17th Annual Conference of the Cognitive Science Society*, pages 660–665, 1995.

[4] G. Grefenstette. SEXTANT: extracting semantics from raw text. *Heuristics*, 1993.

[5] D. Lin. Automatic retrieval and clustering of similar words. In *Proc. of COLING-ACL*, 1998.

[6] J. R. Curran and M. Moens. Improvements in automatic thesaurus extraction. In *Proc. of the ACL Workshop on Unsupervised Lexical Acquisition*, pages 59–66, 2002.

[7] A. Kilgarriff and H. Schuetze. Introduction to the special issue of computational linguistics on the web as a corpus. *Computational Linguistics*, September 2003.

[8] F. Keller and M. Lapata. Using the Web to obtain frequencies for unseen bigrams. *Computational Linguistics*, 29(3), 2003.

[9] M. Poesio. Associative descriptions and salience. In *Proc. of the EACL Workshop on Computational Treatments of Anaphora*, Budapest, 2003.

[10] M. A. Hearst. Automated discovery of wordnet relations. In C. Fellbaum, editor, *WordNet: An Electronic Lexical Database*. MIT Press, 1998.

[11] C. D. Manning and H. Schuetze. *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.

[12] W. A. Woods. What's in a link: Foundations for semantic networks. In Daniel G. Bobrow and Alan M. Collins, editors, *Representation and Understanding: Studies in Cognitive Science*, pages 35–82. Academic Press, New York, 1975. Also appears in [23].

[13] M. Poesio, T. Ishikawa, S. Schulte im Walde, and R. Vieira. Acquiring lexical knowledge for anaphora resolution. In *Proc. of the 3rd LREC*, Las Palmas, Canaria, 2002.

[14] M. Berland and E. Charniak. Finding parts in very large corpora. In *Proc. of the 37th ACL*, pages 57–64, University of Maryland, 1999.

[15] N. Guarino. Concepts, attributes and arbitrary relations. *Data and Knowledge Engineering*, 8:249–261, 1992.

[16] A. Almuhareb and M. Poesio. Attribute- and value-based clustering of concepts. In *Proc. of EMNLP*, Barcelona, July 2004.

[17] D. H. Fisher. Knowledge acquisition via incremental conceptual clustering. *Machine Learning*, 2:139–172, 1987.

[18] G. Karypis. CLUTO: A clustering toolkit. Technical Report 02-017, University of Minnesota, 2002. Available at http://www-users.cs.umn.edu/ karypis/cluto/.

[19] V. Hatzivassiloglou and K. McKeown. Towards the automatic identification of adjectival scales: clustering adjectives according to meaning. In *Proc. of the 31st Annual Meeting of the Association for Computational Linguistics*, pages 172–182, Ohio State University, 1993.

[20] C. Fellbaum, editor. *WordNet: An electronic lexical database*. The MIT Press, 1998.

[21] K. J. Miller. Modifiers in WordNet. In C. Fellbaum, editor, *WordNet*, chapter 2, pages 47–67. MIT Press, 1998.

[22] D. Gross, U. Fischer, and G. A. Miller. The organization of adjectival meanings. *Journal of Memory and Language*, 28:92–106, 1989.

[23] Ronald J. Brachman and Hector J. Levesque, editors. *Readings in Knowledge Representation*. Morgan Kaufmann, San Mateo, California, 1985.