Case-Based Similarity for Social Robots

D.B. Skillicorn[1], R. Billingsley[2], P. Peppas[2], P. Gärdenfors[3], H. Prade[4], and M.-A. Williams[2]

[1] Queen's University
[2] University of Technology Sydney
[3] Lund University
[4] Université Paul Sabatier

Abstract. Social robots must operate in a world of cases that are richer and more varied than those used in conventional case-based systems, since they must be able to represent their internal state, plus their environment, plus the states of humans around them. The attributes describing this environment are qualitative, with multivariate dependencies among them, and at any particular moment those attributes that are relevant will typically form a small subset of all possible attributes.

We design and implement a case-based system to deal with this complexity. It is characterized by (a) decomposition of attributes into subtypes that have qualitatively different similarity structures (in particular, affordances, contexts, properties), and (b) an inference technique for cases that is automatically derived from natural-language descriptions so that a robot can infer it from what it sees (e.g. signage, menus) and (over)hears, rather than being limited to a pre-built ontology. We demonstrate the effectiveness of this system by generating subtype clusters with high semantic relevance.

Classical decision theory assumes that choices depend only on the decision maker's current utilities and probability estimates. However, being able to transfer existing knowledge to new settings is a powerful approach to autonomous activity, for humans, software, and robots. The best known model for explaining memory-dependent decisions is provided by Gilboa and Schmeidler's [8] case-based decision theory, which starts from the assumption that peoples' decisions, are dependent on their previous choices and their memories of the outcomes. The model is not based on probabilities but instead calculates the utility of an alternative weighted by its similarity to previous outcomes.

A case-based reasoning system embodies this idea and must contain (a) a case generation mechanism, (b) a case representation language, (c) a case base of known cases and their outcomes, (d) a measure of similarity between any pair of cases, and (e) a process for comparing a new case to those in the case base to find the one(s) with the greatest similarity measure value [3].

A social robot can use a case-based reasoning system to decide on appropriate actions [6]. The information that defines a case includes data about the robot's own internal state (e.g. level of charge), its environment and, particularly for social robots, the states of the humans with which it is interacting. Cases are therefore high dimensional and intricately structured.

Case representation and case similarity have been well studied, but existing frameworks do not handle the complexity of the kinds of cases social robots must handle. The contribution of this paper is:

- An automatic case generation system derived from natural language, so that the case base can be inferred (and updated) based on language seen or heard by the robot.
- A representation scheme for cases with several novel properties:
  - The attributes of cases are divided functionally into subtypes with qualitatively different similarity structures: affordances, contexts, and properties;
  - Cases can have associated priors, either automatically generated or altered manually;
  - The subtype mechanism is extensible to other kinds of similarity (e.g. between human emotions) using the same framework.
- A method for case generalisation based on spreading activation across the representation system.
- A case similarity measure that allows for considerable precomputation (within each subtype) and that exploits the sparse representation of cases so that the critical path similarity computation is relatively cheap.

Treating the problem in a modular way increases its intelligibility.

1 Case generation and case-case similarity

A case is described by the values of a number of attributes. For example, a robot might be inside or outside; the weather might be sunny; the temperature might be cold; and there might be food in the vicinity. Even this tiny example exposes some of the complexities, however. If the robot is inside, then the weather is less relevant; if there is no human present, then food is less relevant; and temperature might be relevant if it is close to the upper or lower bound of the operating characteristics of the robot, but not otherwise.

A recent summary of case-based reasoning work is Goel and Diaz-Agudo [9]. From their bibliography it is evident that theoretical work has decreased in the past decade, with a much larger number of applied papers in a range of domains. A good summary of the technical issues is the book by Richter and Weber [14].

There have been a number of attempts to generate cases automatically from textual descriptions. For example, Asiimwe *et al.* [1, 2] extract cases from semi-structured descriptions, but these are limited to a particular health-care setting, and require interaction by an expert to make them operational. Asiimwe states "Effective case retrieval and query interpretation is only possible if a domain-specific conceptual model is available and if the different meanings that a word can take can be recognised in the text" [2, abstract]. As we shall show, this is too pessimistic. Others have used web data, especially structured web data such as Freebase and its successors [13]. Data describing processes (e.g. recipes) has also been used a source of cases [5].

Computing the similarity between two cases is a complex problem. Cases are often described by a vector of attribute properties. For any given case, the values of many of these properties will be 'don't care', but the values of others might be critically important. Neither irrelevance nor importance are context-free properties of an attribute in isolation; rather, these depend on the *values* of other attributes. For example, if a robot is inside then the weather is mostly irrelevant, but if it is outside, that same weather is important, perhaps critically important. Thus case similarity requires a complex calculus of how attributes alter the importance of other attributes just to determine the dimensions in which similarity needs to be calculated. However tempting, case similarity cannot be framed simply in terms of a metric on a vector space.

Techniques such as projection to lower dimension using principal component analysis or singular value decomposition [10], or weighting attributes have been tried. In Gärdenfors [7], contexts determine a set of "attention values" that determine the weights of the different properties

An alternative to projection is to select, for each pair of cases, which dimensions should be used to assess their mutual similarity. The difficulty then is to normalize for the dimensions used in the metric. For example, if Case A and Case B are compared using one subset of attributes, while Case B and Case C are compared using another subset, how are the resulting scores to be compared? k-nearest-neighbor techniques have been tried as a way to compensate for some of these issues, but they are sensitive to the choice of k, and to the underlying metric used for distance.

A further problem is that determining which case in the case base is most similar to a given one requires similarity calculations linear in the size of the case base, which is typically large. Attempts to deal with this complexity have used offline clustering of cases [15]. Given a new case, the cluster to which it is most similar is determined first, and then it is compared only to members of that cluster. Others have attempted to compute similarity using a geometric structure such as a k-d tree, but these have an often ignored exponential complexity in the dimensionality.

Although case-based similarity has been extensively studied, it cannot be thought of as a solved problem. We thus design and implement a new approach to similarity intended for, but not limited to, the operation of social robots.

2 Conceptual domains

We begin with the premise that attributes are of qualitatively different kinds. Our strategy is to compare two cases by treating similarity *within* one subtype of attribute differently from similarity

across different kinds of attributes. For example, a bedroom is similar to an office in ways that reflect that they are both rooms, so that they are both inside, unaffected directly by weather, have walls, ceilings, and electricity; but they may or may not have similar temperatures, and this is qualitatively different from their 'structural' properties. An airport passenger terminal is similar to other kinds of rooms (inside, walls, electricity) but can contain some kinds of rooms (offices) but not other kinds (bedrooms) – although an airport passenger terminal can contain a hotel that contains bedrooms.

We define *affordances* to be case attributes associated with what can be done in a particular setting. The interactions between actions is complex – a location in which running is possible is also usually one in which walking is possible (but not necessarily the converse); a location where eating is possible is almost inevitably one where drinking is also possible.

We define *contexts* to be case attributes associated with geography, space, and time, that is they generalize the notion of 'place'. They include places at all granularities: countries, cities, locations such as train stations, airports, shopping malls; and finer-grained locations such as rooms and perhaps even areas within rooms. (Interestingly, humans appear to code across this entire range of granularity using a single mechanism, *place* neurons, each of which appears to represent one 'place' in this general sense [12].) Although there is a natural sense of hierarchy in these examples, the implications for case similarity do not necessarily match this hierarchy. For example, a difference in country (a high-level difference) directly impacts which side of the pavement to walk on (a quite low-level difference).

We define *properties* to be case attributes associated with facts that are true about contexts: temperature, colour, size [7]. The similarity between properties behaves quite differently from the similarity between contexts. For example 'hot' and 'cold' are similar in a linguistic sense, but they are opposites in a semantic sense.

The similarity relationships within each of these subtypes will be different because of their semantics. We use a weighted graph to represent each similarity, with edge weights representing the intensity of (local) similarity between two entities of the same kind (affordances, contexts, or properties). We assume that the similarity structure in each of these domains is static, that is independent of the particular operation of a particular social robot, so they can be completely constructed offline.

Similarity *across* domains is captured by two matrices: affordances $\times$ contexts, and contexts $\times$ properties, whose entries capture the relationship between an object from one domain with an object of the other. This comes in two (overlapping) forms: the generic similarities between domains representing general knowledge and acting as a kind of prior; and the specific similarities associated with each case.

2.1 Domain generation

We generate each domain from a corpus of documents that embody, at least implicitly, relationships and connections among different domain elements. We use the complete simple English Wikipedia corpus, treating each paragraph as a single document, and connect domain elements when they co-occur in a paragraph. The result is perhaps slightly more cerebral than a social robot might need, because Wikipedia has substantial historical and philosophical content, as well as 'general knowledge'.

Contexts capture 'place' aspects of an environment. The difficulty with modelling contexts is that they reflect a wide range of levels of abstraction, from the idea of a country down to a particular location within a room; as well as particular kinds of locations, such as a field, a street, a shopping mall, or a station. Contexts do not interlock cleanly. A coffee shop may be *inside* a shopping mall, station, or airport; but also outside on a street, where there might also be a coffee cart. There is little hope for a comprehensive, hierarchical taxonomy or ontology of contexts, even if the resources to construct it were available.

We represent contexts using a network – contexts are nodes and weighted edges reflect the similarity (or overlap) between contexts. We are agnostic about whether similarity reflects horizontal relationships or hierarchical ones.

An initial network of contexts in constructed as follows:

1. Label the corpus with parts of speech, construct a document-word matrix, retain words that are sufficiently frequent globally, and select those columns associated with *nouns*. Because we process

the corpus as paragraphs, we do not row normalize the matrix (i.e. assuming that all documents are about the same length).
2. Compute the word-word similarity matrix (i.e. dot product of the columns) Interpret this matrix as the adjacency matrix of the network of context-context relationships.

This network typically has a single, large connected component, and then a large number of very small components, mostly pairs. No attempt is made to deal with hierarchical contexts explicitly; it is assumed that these will be induced by the co-occurrence of the relevant terms in documents. Thus the co-occurrence of 'desk' and 'office' reflects one kind of place similarity, while the co-occurrence of 'office' and 'building' reflects another.

This network suffers from the problem that common (and often quite general) nouns co-occur with many other nouns. These common nouns distort the relationships between every other pair of nouns making them seem more similar than they 'should be'. Common nouns also tend to co-occur with other common nouns, worsening the problem. The solution is to build a metric on the space of words that discounts the global popularity of words appropriately. We use an approach developed by Brand [4]. It uses the following steps:

1. Rather than computing word-word similarity based on distances from the adjacency matrix, calculate the matrix of pairwise (square roots of) commute distances between each pair of words;
2. Calculate the cosine similarity of the commute distance embeddings and use this as the metric for word-word similarity.

When the adjacency matrix is symmetric with zero diagonal, the commute distances do not need to be explicitly computed – the matrix of cosine similarities of commute distances can be computed directly as the Moore-Penrose inverse of the combinatorial Laplacian of the adjacency matrix. Matlab code is shown in the appendix. Using the cosine distances implicitly projects onto a unit hypersphere, and so discounts the effect of well-connected nodes on the set of pairwise commute distances.

Although the adjacency matrix is typically quite sparse, the cosine similarity matrix is dense. It seems natural to make it sparse again by removing smaller values. Maier *et al.* [11] have shown that the most effective way to do this in general is to keep either symmetric or mutual k-nearest neighbors, but k should be a significant fraction of the matrix size (and certainly more than logarithmic). This strategy does not work well in our setting because of properties of language – common words have more, weaker neighbors in the dense network than rarer words do, which causes complex distortions. For the purposes of illustration, we keep the top k values in each row of the context-context matrix (and also the affordance-affordance matrix), for k around 20.

The role of this network of pairwise similarities is to allow generalization across place. A case with the context of a bedroom is likely to have some similarity with a case involving an office, a bathroom, or a kitchen. The network is used to spread activations (generalise contexts) in a meaningful way, so that related places become relevant in the structure of each case, and the calculation of similarity between cases.

Properties are not related in the same way as contexts, and we treat them differently. An example will illustrate the issue: the similarity between 'hot' and 'cold' is complex. They are similar in the sense that they represent different regions of the same dimension ('temperature'). They are also similar in a lexical sense since they tend to co-occur with the same words ('hot day', 'cold day') and so corpus-derived measures will tend to find them similar. However, semantically they are dissimilar, in fact opposites. Property similarity requires a subtle balancing of lexical similarity, which tends to associate terms that can fill the same 'slot', with semantic dissimilarity, terms whose meanings are different.

We design a similarity mechanism that captures aspects of both lexical and semantic similarity by connecting properties in a tree. The leaves of this tree are the basic properties, and the internal nodes are constructed 'superproperties' which capture generalized similarity (but only *indirectly*).

As before we construct the property tree based on a corpus, selecting adjectives as the signals of properties. The process is as follows:

1. Given a corpus, construct a document-word matrix (as for contexts), select the most frequent words, and select those columns associated with adjectives and gerunds.
2. Compute the word-word similarity matrix (i.e. dot product of the columns)
3. Convert this network into a tree by repeating these steps:
   - select the two most similar nodes and create a new supernode labelled by concatenating the labels of the original two nodes;
   - move any edges that connected to the selected nodes so that they now attach to the new supernode;
   - reweight these edges to the mean of the weights that connected them to the original, selected nodes and reduce the weights connecting the two selected nodes to the new supernode so that they will never be selected again (guaranteeing a tree).

The tree (really a forest) typically has one large connected component, with a number of much smaller components.

This mechanism also associates, indirectly, adjectives that do not convey the same kind of property but frequently co-occur in phrases, for example "nice sunny (day)" and "deep blue (sea)".

Affordances capture the actions possible in a setting. Like contexts, they are related in ways that are connected both hierarchically ('lie' typically also implies the affordance of 'sit') and horizontally ('eat' implies the affordance of 'drink'). We infer affordances from *verbs* (but not auxiliary verbs such as 'be' and 'have') and construct a network whose nodes are verbs and whose weighted edges reflect co-occurrence in paragraphs of the corpus in the same way as for contexts. As before, we compute an affordance-affordance similarity matrix based on the cosine differences of the commute distances of this network. We also threshold the entries using the same strategy as for the context-context matrix.

These three network structures represent background or basic similarity within each of the three domains: which affordances are similar to one another, which contexts are similar, and which properties are similar.

Connections between domains are two non-symmetric matrices connecting affordances to contexts, and contexts to properties. The matrix connecting affordances and contexts has one row for each verb and one column for each noun. The matrix connecting contexts and properties has one row for each noun and one column for each adjective.

We use these matrices in two ways. First, such matrices can be thought of as general knowledge priors on the relationships between contexts and properties, and affordances and contexts. We compute these cross-domain matrices from the co-occurrences of verbs with nouns, and nouns with adjectives, respectively, in the paragraphs of the corpus. Thus they are verb-noun and noun-adjective co-occurrence matrices. It is possible to convert these matrix entries into more meaningful associations, using the Brand transformation, but it is a great deal more complex for non-symmetric, non-square matrices. Instead, we threshold them by setting correlations below the mean to zero, and convert them to binary matrices (so that they contain a 1 when a particular verb-noun or noun-adjective pair occurs more often than the global average in the corpus). We use these binary matrices to generalize cases where one of the domains is not mentioned.

Second, we use matrices of this form to represent each case: a case *is* the weighted co-occurrences of affordances, contexts, and properties; and case-case similarity is a metric on pairs of matrices.

Thus our model has five static, precomputed pieces: the context and affordances nets, the property tree, and the affordance $\times$ context, and context $\times$ property cross matrices. All five can be thought of as priors on the world, extracted from textual descriptions about it.

Each case, or set of cases, is also represented by a pair of affordance $\times$ context, and context $\times$ property cross matrices, which will typically be much sparser than the prior versions.

The overall structure of the representation is shown in Figure 1.

2.2 Case generation and representation

The simplest case is a triple of ⟨affordance, context, property⟩ derived from or based on a piece of text. For example, a social robot might see an ad "Buy iced coffee from coffeechain" or might be asked by

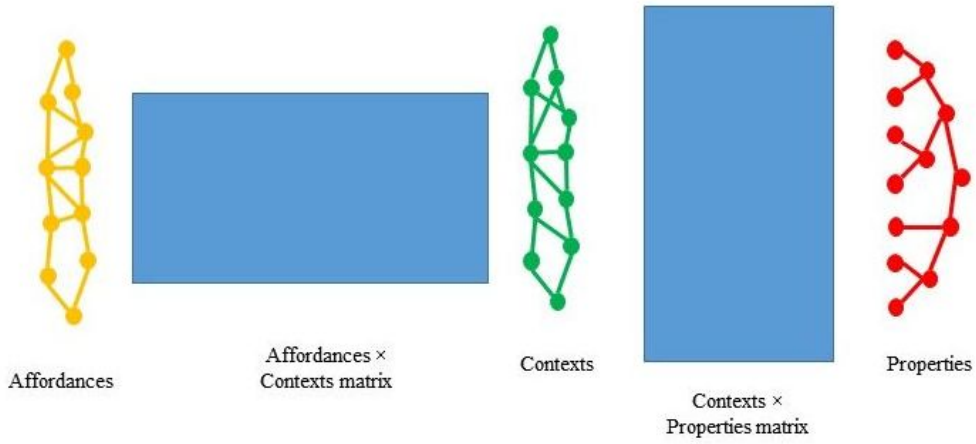

Fig. 1. Overall structure of the representation of similarity

a human "Buy me an iced coffee from coffeechain". Both generate a triple involving the affordance 'buy', the property 'iced' and the context 'coffeechain'.

The simplest representation of this case has a 1 in the 'coffeechain' row and 'iced' column of the contexts × properties matrix, and a 1 in the 'buy' row and 'coffeechain' column of the affordances × contexts matrix.

However, this representation doesn't leverage any of the similarity information captured in the three affordance, context, and property structures. We generalise each case, using a spreading activation intuition. The matrices associating affordances and contexts, and contexts and properties, are then enhanced to reflect the newly activated nodes. The spreading activation is shown in Figure 2. The solid lines are the relationships among concepts precomputed from the underlying corpus. The dashed lines represent the properties of the particular case. The dotted lines are relationships that are inferred because of the interaction of domain and case. The dashed and dotted lines correspond to entries that will be inserted into the affordance × concept and concept × property matrices to describe this case.

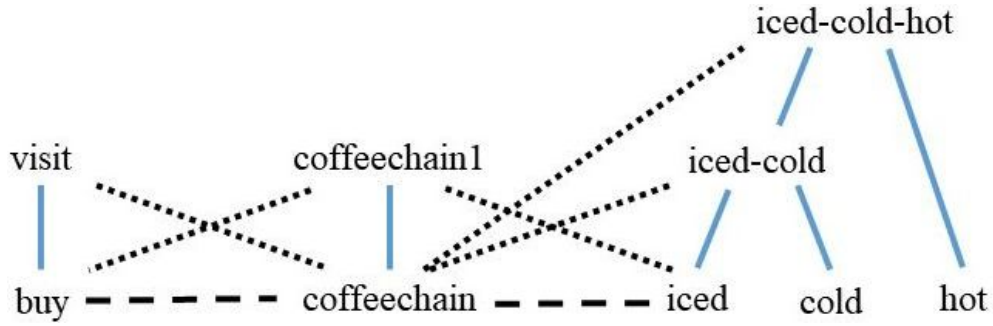

Fig. 2. Spreading activation from the case buy-coffeechain-iced

We can extend this simple case, a single triple, to a set of triples with the entries in the matrices extended to include all of the weighted edges generated from them. A case can even be constructed automatically from a piece of text by processing it to extract the co-occurrences it contains.

But what if the case contains only a pair of elements, rather than a triple? Here we can use the default, prior versions of each of the matrices to infer the missing associations. For example, if we begin with the triple 'buy' and 'iced', we can find all of the contexts that have a non-zero connection to 'buy' and create the affordances × contexts matrix that has non-zero edges (summing to 1) for all of these connections. This reflects the fact that 'buy' is associated with many contexts, we don't know which one, but we can activate all that make sense.

Similarly, we find all of the contexts that have non-zero connections to 'iced' and create a context × properties matrix that has non-zero entries (summing to 1) for all of these connections. Again this reflects all of the contexts where 'iced' is a possible property.

When the 'missing' element of a triple is either a property or an affordance, an alternative is simply not to create any edges in the relevant matrix. (This reflects the fact that contexts bind together both affordances and properties. It would also be possible to define, and build, an affordances × properties matrix using exactly the same strategy but ambiguities abound, and contexts usually resolve them. For example, 'run cold' could mean water from a tap, or blood metaphorically, or a car engine – it is the context word which separates the possible similarities between the affordance 'run' and the property 'cold').

2.3 Case-case similarity

The measure of case similarity is the sum of the Frobenius norm differences between the corresponding matrices for the two cases – how similar are they with respect to their affordances × contexts matrices, and their context × property matrices. Although these matrices are potentially large, they are very sparse (since even the most complex case involves only a relatively small number of affordances, contexts, and properties) so the calculation of the Frobenius norms is inexpensive. An implementation of the case-based reasoning system has been built in Matlab.

3 Examples

We first show how the affordance, context, and property matrices infer similarities from the corpus.

Table 1 shows the nearest neighbours of some of the common affordances. Similarity between verbs, especially common verbs, is quite weak; but it is encouraging that different parts of the same verb are detected as similar.

Table 1. Example of affordance neighbours

Base word	Ranked near neighbours
born	died, retired, known
died	born, united, retired
make	used, promote, use, get, made
use	used, make, get, united, using, made, called
publish	include, becomes, published, played, help

Table 2 shows the nearest neighbours of some contexts. These are quite compelling, given that they were generated automatically. Note especially the words related to 'dylan'.

Table 2. Examples of context neighbours

Base word	Ranked near neighbours
people	census, things, population
world	championship, heritage, sites, war
war	world, coalition, rebellion, succession
music	records, lyrics, recordings, composers
football	team, league, player, confederation
france	department, region, commune
america	europe, squad, football, states
dylan	bob, singer, footballer, actor, actress, thomas

Property adjacencies have a different structure. Each of the supernodes contains its substructure because the adjectives it is built from are contained in its name, and the order gives the order of accretion. Table 3 shows some superproperties. These are compelling given that they are automatically generated from the corpus.

We now illustrate a complete case, based on the triple 'sing-dylan-musical'. The related terms and their associated weights are shown in Table 4. The entry of the affordance-context matrix at the row corresponding to 'singing' and the column corresponding to 'dylan' will be 0.029, the entry at the row

Table 3. Example of property supernodes, and their components in the order in which they are generated

Supernodes	
late-early	minor-major
catholic-roman-holy	brown-dark
greek-ancient	secondary-primary
chartered-worshipful	economic-political
bad-good	cold-hot
negative-positive	lowest-highest
fricative-voiceless	orthodox-russian-ukrainian
middle-high-elementary	false-true
left-right	make-female
short-long	larger-smaller
private-public	atomic-nuclear
black-white-red-yellow-green-blue-big	
golden-best-original-outstanding-musical-nominated-special	

corresponding to 'sing' and column 'bob' will be 0.009, and so on. Recall that the entry corresponding to 'singing' and 'bob' remains zero. In other words only entries in a single row and in a single column may become non-zero.

Table 4. Similar domain elements for the case 'sing-dylan-musical'

Affordances	Weights	Contexts	Weights	Properties	Weights
sing	1.0	dylan	1.0	musical	1.0
singing	0.029	bob	0.009	outstanding-musical	1/2
sings	0.022	singer	0.005	golden-best-original-outstanding-musical	1/3
publish	0.019	footballer	0.005	+ nominated	1/4
play	0.016	actor	0.005	+ special	1/5

Suppose that we only have the case 'dylan-musical'? Then all of the affordances with a non-zero entry in the column for 'dylan' in the affordance-context cross matrix are selected and copied into the case matrix, but downweighted so that their total weight sums to 1. Thus we activate all affordances that have co-occurred in the corpus with 'dylan', but with low weights to reflect uncertainty about which, if any, might make sense.

The set of affordances that are associated with 'dylan' are shown in Table 5. These are mostly verbs associated with either music or theatre; even the odd one – 'massacre' – makes sense because one of the shooters in the Columbine school shooting was named Dylan.

Table 5. All affordances activated by the context 'dylan'

born, united, known, called, released, named, get, live, written, play, died, directed, acting, love, featuring, inspired, programming, mixing, massacre

Similarly, if no property is given, then the set of potentially relevant properties can be found by choosing all of the adjectives that co-occur frequently with 'dylan'. These are shown in Table 6. Many are nationalities, typically ones where Bob Dylan is a celebrity (Dylan Thomas is Welsh).

4 Discussion and conclusions

As robots increasingly require context to make their decisions in situations of varying familiarity, case-based approaches seem increasingly appropriate to determine which facets of their environment are relevant. We have designed a system for case-based similarity, including a range of techniques suitable for the distinctive nature of social robot situational representation, that differs from existing approaches because:

Table 6. All properties activated by the context 'dylan'

american, new, different, british, english, french, german, famous, japanese, italian, best, big, canadian, welsh, assistant

- It divides attributes into subsets with characteristically different properties; in particular, with different kinds of mutual similarity;
- It infers the similarity structure of each subtype using word co-occurrences in a natural language corpus, enabling domains (subtypes and their similarities) to be built automatically (rather than requiring experts);
- Each domain's similarity structure is precomputed, reducing the cost of case construction, and the cost of case-case comparison (because cases share underlying domain structures);
- It uses properties of domains to generalize a given case, beyond the given data, but in a semantically consistent way;
- It represents cases in a way that reflects their inherent sparseness, reducing the cost of case bases;
- It makes fast case-case comparisons possible because of the sparse representation of each.

Matlab code: Brand transformation to remove common nodes

```
g = words' * words; % co-occurrence matrix
g(1:numwds+1:numwds*numwds) = 0; % zero the diagonal
coli = ones(numwds,1);
almostp = diag(g*coli) - g; % combinatorial Laplacian
p = pinv(almostp);  % Moore-Penrose inverse - cosine diffs of commute distances
d = sqrt(inv(diag(diag((p)))));  % normalize
g = d * p * d;
```

References

1. S. Asiimwe, S. Craw, B. Taylor, and N. Wiratunga. Case authoring: From textual reports to knowledge-rich cases. In *International Conference on Case-Based Reasoning, ICCBR 2007, Springer LNAI4626*, pages 179–193, 2007.
2. S.M. Assiimwe. *A Knowledge Acquisition Tool to Assist Case Authoring from Texts*. PhD thesis, Robert Gordon University, 2009.
3. R. Billingsley, H. Prade, G. Richard, and M-A. Williams. Towards analogy-based decision - a proposal. In *12th International Conference on Flexible Query Answering Systems*, pages 28–35, 2017.
4. M. Brand. A random walks perspective on maximizing satisfaction and profit. In *SIAM Conference on Optimization*, May 2005.
5. V. Dufour-Lussier and J. Lieber. Evaluating a textual adaptation system. In *International Conference on Case-Based Reasoning*, 2015.
6. T. Fitzgerald and A. Goel. A case-based approach to imitation learning in robotic agents. In *International Conference on Case Based Reasoning*, page 10pp, 2014.
7. P. Gärdenfors. *Geometry of Meaning: Semantics Based on Conceptual Spaces*. MIT Press, 2014.
8. I Gilboa and D. Schmeidler. *A Theory of Case-based Decisions*. Cambridge University Press, 2001.
9. A.K. Goel and B. Diaz-Agudo. What's hot in case-based reasoning. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)*, pages 5067–5069, 2017.
10. G.H. Golub and C.F. van Loan. *Matrix Computations*. Johns Hopkins University Press, 3rd edition, 1996.
11. M. Maier, M. Hein, and U. von Luxburg. Optimal construction of k-nearest neighbor graphs for identifying noisy clusters. *Theoretical Computer Science*, 40(19):1749–1764, 2009.
12. J. O'Keefe and N. Burgess. Geometric determinants of the place fields of hippocampal neurons. *Nature*, 381(6581):425–428, 1996.
13. J.A. Recio-García, M.A. Casado-Hernández, and B. Díaz-Agudo. Extending CBR with multiple knowledge sources from web. In *International Conference on Case based Reasoning*, pages 287–301, 2010.
14. M.M. Richter and R.O. Weber. *Case Based Reasoning*. Springer, 2013.
15. Q. Yang and J. Wu. Enhancing the effectiveness of interactive case-based reasoning with clustering and decision forests. *Applied Intelligence*, 14(1):49–64, January 2001.