Inducing semantic relations from conceptual spaces: a data-driven approach to plausible reasoning

Joaquín Derrac, Steven Schockaert

Cardiff University, School of Computer Science & Informatics, 5 The Parade, Cardiff CF24 3AA, UK

Abstract

Commonsense reasoning patterns such as interpolation and a fortiori inference have proven useful for dealing with gaps in structured knowledge bases. An important difficulty in applying these reasoning patterns in practice is that they rely on fine-grained knowledge of how different concepts and entities are semantically related. In this paper, we show how the required semantic relations can be learned from a large collection of text documents. To this end, we first induce a conceptual space from the text documents, using multi-dimensional scaling. We then rely on the key insight that the required semantic relations correspond to qualitative spatial relations in this conceptual space. Among others, in an entirely unsupervised way, we identify salient directions in the conceptual space which correspond to interpretable relative properties such as 'more fruity than' (in a space of wines), resulting in a symbolic and interpretable representation of the conceptual space. To evaluate the quality of our semantic relations, we show how they can be exploited by a number of commonsense reasoning based classifiers. We experimentally show that these classifiers can outperform standard approaches, while being able to provide intuitive explanations of classification decisions. A number of crowdsourcing experiments provide further insights into the nature of the extracted semantic relations.

Keywords: Conceptual spaces, Dimensionality reduction, Qualitative spatial relations, Commonsense reasoning

Email addresses: j.derrac@cs.cardiff.ac.uk (Joaquín Derrac), s.schockaert@cs.cardiff.ac.uk (Steven Schockaert)

1. Introduction

Structured knowledge bases are becoming increasingly important in applications such as question answering, semantic search and recognizing textual entailment. Applying logic based methods in such applications is challenging, however, because relevant knowledge is often not available in a symbolic form [1]. As a result, several authors have recently looked at techniques for automatically extending popular knowledge bases, such as DBpedia[1], YAGO[2], Freebase[3] and ConceptNet[4]. One possibility is to use external knowledge [2], and for example rely on information extraction techniques to fill in missing values for prominent attributes (e.g. missing birth dates). Other approaches rely on exploiting regularities within the knowledge base, e.g. by learning rules capturing probabilistic dependencies [3] or using matrix factorisation [4]. A third class of approaches relies on commonsense reasoning, inspired by the observation that humans have a remarkable ability to cope with missing knowledge, by drawing plausible but unsound conclusions when their knowledge is insufficient to answer a given question [5]. Most existing approaches in this class rely on similarity based reasoning [6, 7, 8], i.e. on the assumption that similar concepts tend to have similar properties:

Similarity based reasoning if we know that Alice enjoyed the Lord of the Rings trilogy, we can derive that she will probably like the Hobbit trilogy as well, as both trilogies are quite similar.

The required similarity degrees are often obtained from so-called distributional models, i.e. from the co-occurrence patterns of the corresponding natural language terms in large text collections. The popularity of similarity based methods can be largely explained by the relative ease with which such distributional models can be learned. However, similarity based reasoning also has two important limitations. First, it can only be used when there are sufficiently similar concepts that we can exploit (e.g. if we do not know whether Alice liked the Lord of the Rings trilogy, it would be much harder to use similarity based reasoning for predicting whether she would like the Hobbit, as there are few other films that are similar to it). Second, similarity

[1] http://dbpedia.org/About

[2] `http://www.mpi-inf.mpg.de/yago-naga/yago/`

[3] `http://www.freebase.com`

[4] `http://conceptnet5.media.mit.edu`

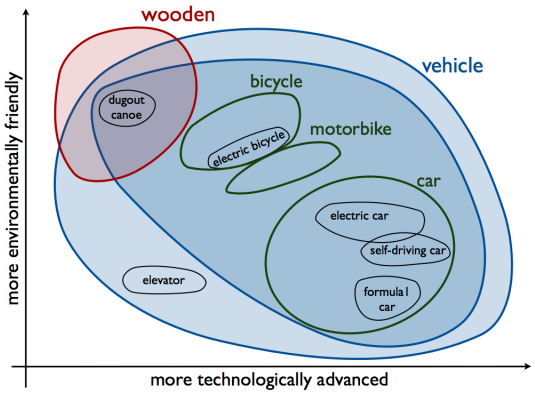


Figure 1: A conceptual space of vehicles.

degrees are highly context-dependent (e.g. red and white Burgundy wine are similar in some sense, but they should be paired with very different types of food). To alleviate these limitations, we propose to augment similarity based reasoning with two other patterns of commonsense reasoning:

Interpolative reasoning if we know that undergraduate students and PhD students are both exempt from paying council tax in the UK, we can plausibly conclude that master's students are also exempt from paying this tax, given that master's students are conceptually between undergraduate students and PhD students.

A fortiori reasoning if we know that buying beer is illegal under the age of 18 in the UK, we can plausibly derive that buying whiskey is also illegal under the age of 18, since whiskey is stronger than beer.

Unfortunately, the semantic relations that are needed to automate these forms of commonsense reasoning are not commonly available. Large-scale semantic knowledge bases such as DBpedia, YAGO and Freebase mainly encode attributional knowledge such as "Chianti is made from the Sangiovese grape", while we need relational knowledge such as "Chianti is generally less tannic than Cabernet Sauvignon". Lexical resources such as WordNet[5] and ConceptNet do contain relational information, but they are limited to a small set of predefined relations (e.g. synonyms and is-a relations).

[5] `http://wordnet.princeton.edu`

In this paper, we will show how the required semantic relations can be obtained by interpreting them as qualitative spatial relations in a particular kind of distributional model. Specifically, we will obtain semantic relations from *conceptual spaces* that have been induced from text corpora. Conceptual spaces [9] are metric spaces which are used to encode the meaning of natural language concepts and properties. In most applications, conceptual spaces are assumed to be Euclidean. They are typically high-dimensional, with each dimension corresponding to a primitive cognitive feature. Specific entities then correspond to points in the conceptual space, while natural concepts and properties are posited to correspond to convex regions [9]. Figure 1 shows a simple example of a two-dimensional conceptual space of vehicles, although it should be noted that most conceptual spaces will have a considerably higher number of dimensions. An important observation is that many types of semantic relations between vehicles correspond to qualitative spatial relations in this conceptual space. For example, the semantic is-a relationship corresponds to a spatial part-of relationship (e.g. the region for bicycle is included in the region for vehicle, because every bicycle is also a vehicle). Furthermore, we can identify conceptual betweenness with geometric betweenness (e.g. the region for motorbike is geometrically between the regions for bicycle and car, and accordingly the properties of a motorbike can be thought of as being intermediate between those of a bicycle and those of a car). Vagueness can be modelled by modelling concepts as fuzzy sets, or more simply, as nested sets of convex regions (e.g. elevators can be considered as borderline cases of vehicles). Finally, relative properties such as "more technologically advanced" correspond to direction relations: the more a vehicle is located to the right, the more it is technologically advanced.

The aim of this paper is to investigate (i) how suitable conceptual spaces can be induced from large text corpora, (ii) how interpretable semantic relations can be derived from these conceptual spaces, and (iii) how these relations can be used to learn categorization rules based on the aforementioned commonsense reasoning patterns. Compared to standard machine learning approaches, these categorization rules will have the advantage that they allow us to produce intuitive justifications for inferred facts, which we believe is paramount in applications that rely on imperfect reasoning. Many current recommender systems, for example, essentially use some form of similarity based reasoning, which allows them to provide explanations of the form "we think that you will like X because you have previously expressed an interest in Y and Z". Similarly, we could imagine a wine recommendation engine to

provide suggestions such as "while beef dishes are often paired with Cabernet Sauvignon, we recommend a medium-bodied wine such as Chianti for beef carpaccio, because uncooked meat is usually paired with lighter wines than grilled meat", if it has no specific knowledge on what wines to pair with beef carpaccio. As another example, consider the problem of automatically extending knowledge bases such as Freebase. Several methods for automatically extending such knowledge bases have already been proposed [4, 10, 11, 3], which could be useful, among others, for developing semantic search systems on the web of data that go beyond fact retrieval. However, given that no method can provide perfect accuracy, in such applications it becomes crucial to explain to users why a particular answer is believed, allowing them to assess the credibility of inferred facts.

Beyond commonsense reasoning, inducing semantic relations is also useful for applications such as critique based recommendation and search [12]. The idea of critique based systems is to enable the user to find an item of interest through an interactive process. First, a list of options is displayed, based on an initial query (e.g. hotels in Cardiff). Then the user can critique these options, specifying how the desired item differs from a suggested item (e.g. "like this hotel, but cheaper"). Most existing work is limited to domains where the relevant attributes are clearly defined, and the corresponding values are explicitly provided. One exception is [13], which proposes a critique based movie recommender. Using a supervised method, their system allows users to specify, for instance, that they want "a film like this one, but grittier". Similarly, [14] proposes a critique based image search engine, based on a supervised method that learns the degree to which visual attributes apply to images, e.g. "I want to buy shoes like these, but shinier". Clearly, such supervised methods are difficult to scale beyond specific domains. In contrast, the methods we develop in this paper are unsupervised, and could thus enable critique based search in a much broader set of domains.

The remainder of this paper is structured as follows. After reviewing related work in Section 2, Section 3 explains how we use multi-dimensional scaling to derive conceptual spaces from text corpora. Then, in Section 4 we show how interpretable semantic relations can be induced from these conceptual spaces in an entirely unsupervised way. Subsequently, Section 5 discusses how these (mostly qualitative) semantic relations can be used in categorization problems. Finally, experimental results are presented in Section 6. This paper significantly extends our work in [15] and [16].

2. Related work

2.1. Formalizing commonsense reasoning

Similarity based reasoning, i.e. the view that similar concepts tend to have similar properties, has been widely studied. In cognition, it lies at the heart of the prototype and exemplar based models of categorization [17, 18, 19]. In machine learning, it forms the basis of the k-nearest neighbour method [20]. Similarity based reasoning has also been studied in logic. For example, [21] uses a connectionist approach for modelling similarity, and proposes an inference mechanism that combines rule based and similarity based reasoning. Several other approaches to similarity based reasoning rely on fuzzy logic for encoding numerical similarity degrees in a logical framework [22, 23].

One of the challenges in developing a logic based encoding of similarity based reasoning is that similarity degrees tend to be subjective and context-dependent. Moreover, it is unclear how similar two concepts need to be before plausible conclusions can be obtained, which makes it difficult to automate similarity based reasoning in a principled way. Interpolation avoids these issues by taking the qualitative notion of conceptual betweenness as primitive. Interpolative reasoning has been investigated as a technique for completing fuzzy logic rule bases [24]. Recently, a propositional logic that supports interpolation as a form of commonsense reasoning has been proposed in [25]. The idea of interpolation is also similar in spirit to the approach from [26], where similarity degrees are avoided by introducing a ternary modality encoding comparative similarity. Unlike betweenness, however, comparative similarity is not invariant under linear transformations. Rescaling the dimensions of a conceptual space (which is a linear transformation) is a common method for modelling changes in context, which suggests that comparative similarity may be more context-dependent than betweenness. Another approach to avoid similarity degrees is the idea of statistical predicate invention proposed in [27]. Essentially this approach is based on clustering, where a clustering can be seen as a binary similarity relation (i.e. two elements are considered similar iff they belong to the same cluster) and the idea is that entities from the same cluster tend to have the same properties. However, rather than using a single clustering, the approach from [27] relies on multiple clusterings, each reflecting different aspects of the domain. A method is then proposed to learn which types of properties can be induced from which clusterings.

A fortiori reasoning assumes that concepts and properties can be ordered in a natural way and that these orderings are co-monotone. In the example

in the introduction, alcoholic drinks are ordered according to strength and it is assumed that this ordering is co-monotone with the legal drinking age, which allows us to draw conclusions about the legal drinking age for whiskey from knowledge about the legal drinking age for beer. This form of inference has been analysed in [28], where it is proposed as the basis of a method for deriving plausible values for missing features, given a collection of feature vectors. The method proposed in [28] is based on the idea of inducing a partial order from a set of feature vectors, and choosing the missing value such that the resulting partial order is maximally regular in some sense.

Analogical reasoning can be seen as a generalization of a fortiori reasoning, where the ordering on concepts and properties is not explicitly given, but is rather encoded implicitly in the form of examples. Instead of assuming that the natural ordering of concepts and properties is co-monotone, we then assume that concepts which differ in analogous ways have properties which differ in analogous ways. Several authors have studied analogical proportions, i.e. statements of the form "salmon tartare is to grilled salmon what beef carpaccio is to grilled steak", and their use in classification in recent years [29, 30, 31, 32]. Most of these approaches are restricted to binary or nominal attributes, although recently some promising results have been obtained for numerical attributes as well [32]. The use of analogical-proportion based reasoning in logic has been considered in the approach from [25], where the more general notion of extrapolative reasoning was studied. Note that while analogical reasoning is a broad area of study, we will not be concerned with approaches that aim to transfer knowledge from one domain to another [33].

It should be noted that existing approaches to commonsense reasoning typically require that objects are described using well-defined attributes, encoded as binary or real-valued features. In contrast, we consider scenarios where we only have access to text documents describing the entities of interest. Moreover, we are not only interested in identifying that e.g. beer relates to wine like wine relates to whiskey, but also in naming this relationship, i.e. wine is *stronger than* beer, and whiskey is *stronger than* wine.

2.2. Acquiring semantic relations

It is convenient to represent the meaning of terms or documents as points, vectors or regions in a Euclidean space. Such representations are known as vector-space models, conceptual spaces, or semantic spaces, and are popular in areas such as cognition [34], computational linguistics [35, 36, 37],

information retrieval [38, 39] and knowledge representation [9, 4]. In information retrieval, it is common to represent documents as vectors with one component for every term occurring in the corpus. In many other applications (and sometimes in information retrieval) some form of dimensionality reduction is typically used to obtain vectors whose components correspond to concepts. One of the most popular techniques, called latent semantic analysis (LSA [39]), uses singular value decomposition (SVD) to this end. Multi-dimensional scaling (MDS [40]) is another popular method for dimensionality reduction, which builds a vector-space representation from pairwise similarity judgements. It is popular, among others, in cognitive science to interpret pairwise similarity judgements obtained from human assessors.

Most approaches represent natural language terms as points or vectors. One notable exception is the work of Gärdenfors on conceptual spaces [9], where properties and concepts are represented using convex regions, while specific instances of a concept are represented as points. This has a number of important advantages. First, it allows us to distinguish borderline instances of a concept from more prototypical instances, by taking the view that instances which are closer to the center of a region are more typical [9]. A second advantage is that using regions makes it clear whether one concept subsumes another (e.g. every pizzeria is a restaurant), whether two concepts are mutually exclusive (e.g. no restaurant can also be a beach), or whether they are overlapping (e.g. some bars serve wine but not all, some establishments which serve wine are bars but not all). Region based models have been shown to outperform point based models in some natural language processing tasks [41]. On the other hand, using regions is computationally more demanding, and learning accurate region boundaries for a given concept would require a prohibitive amount of data. In this paper, we essentially view point based representations as coarse-grained approximations of conceptual spaces, where points correspond to fine-grained categories instead of specific instances, while convex regions are used to model higher-level categories.

To date, vector-space representations have almost exclusively been used to estimate the similarity between terms (or documents), e.g. to find documents that match a given query in information retrieval, or to find synonyms in computational linguistics. The use of similarity degrees from vector-space models for logical reasoning has been explored in [6]. In particular, the authors extend a Markov logic theory with rules that essentially encode that similar concepts are likely to have the same properties. It is shown that implementing this form of similarity based reasoning improves a system for

recognizing textual entailment. In [7], distributional similarity is used to improve reasoning about commonsense knowledge bases such as ConceptNet. Other authors use similarity based reasoning in a more implicit way for automatically extending (light-weight) knowledge bases. For example, [4] represents ConceptNet as a matrix, with rows corresponding to concepts and columns corresponding to properties, and then applies singular value decomposition on that matrix to identify plausible properties that are missing from ConceptNet. Along similar lines, [10] represents YAGO as a tensor and uses a tensor decomposition method to find plausible properties that are missing from YAGO. Even though these methods do not explicitly use a similarity measure, the assumption underlying the use of dimensionality reduction methods is that similar concepts are likely to have similar properties.

Beyond similarity, a few authors have looked at learning analogical proportions $a : b :: c : d$ from data. If the analogical proportion $a : b :: c : d$ holds, the pairs (a, b) and (c, d) are called relationally similar [42]. To learn relationally similar pairs from a text corpus, in [42] a matrix is compiled with rows corresponding to pairs of words (a, b) and columns corresponding to phrases P. The matrix itself encodes whether the corpus contains sentences in which the phrase P connects the words a and b. For example, in a sentence such as "a kitten is a young cat" the words kitten and cat are connected by the phrase "X is a young Y". Singular value decomposition is then applied to the matrix, after which relationally similar pairs of words can be identified. The method is thus able to learn that e.g. (kitten,cat) is relationally similar to (puppy,dog) by observing that both pairs of terms tend to be connected by similar phrases (e.g. "X is a young Y"). While showing promising results, this approach has the drawback of being computationally demanding. Moreover, it can only discover relational similarity between pairs of words that are mentioned in the same sentence sufficiently often. An alternative method, which does not suffer from these drawbacks, has been proposed in [43]. This method learns two semantic spaces, based on two different notions of similarity, referring respectively to functional role and the domain of terms. In this way, the approach can discover that (carpenter,wood) and (mason,stone) are relationally similar, because: (i) carpenter and wood are terms from the same domain, (ii) mason and stone are from the same domain, (iii) carpenter and mason have a similar function, and (iv) wood and stone have a similar function. However, this approach is not suitable for identifying semantic relations among entities of the same type (e.g. between pairs of movies), which is what we focus on in this paper. A few approaches have tried to learn ana-

logical proportions by identifying (approximate) parallelograms in a learned semantic space, including the approach based on multi-dimensional scaling from [44] and the neural network based approach from [45]. Note that the aforementioned approaches essentially use a geometric representation of the domain of interest to discover analogical proportions. In [46] the opposite problem is discussed: given a set of analogical proportions that are known to hold, it is studied how can we learn a better geometric representation (in the context of visual object categorization).

A rather different line of work uses relation extraction methods to extract semantic relations from natural language sentences. For example, NELL[6] [47] extends and populates an ontology by continuously reading web documents, relying only on minimal human supervision. Along similar lines, the Open Information Extraction project[7] [48] aims to extract semantic relations without specifying the types of semantic relations in advance, again by analysing natural language sentences. SOFIE [49] also extracts semantic relations from natural language, but focuses on extending an existing ontology such as YAGO. Relation extraction from natural language is clearly a promising method to identify properties of entities (e.g. the fact that the Shining movie was released in 1980). However, it is less clear to what extent relation extraction can successfully identify semantic relations between entities of the same type, such as "the Shining is (generally considered) more terrifying than the Hunger games". Indeed, there are few sentences on the web that explicitly compare two movies in such a way[8] (unless in particular cases, such as when a sequel is compared to the original movie).

3. Inducing conceptual spaces from data

In Section 4, we will discuss how semantic relations between entities of the same kind can be induced from conceptual spaces. These relations will then be used in Section 5 as the basis for commonsense reasoning based classifiers. In this section, we first focus on data acquisition, and in particular on how we have induced the conceptual spaces that will be used throughout the paper. We will focus on conceptual spaces in three domains: place types,

[6] `http://rtw.ml.cmu.edu/rtw/`

[7] `http://ai.cs.washington.edu/projects/open-information-extraction`

[8] For instance, a google search for "the shining is * than the hunger games" yields no results (verified on 1 April 2015).

movies, and wines. Sections 3.1–3.3 explain how we have compiled a text corpus about entities in these domains. Section 3.4 then explains how we use multi-dimensional scaling to obtain a conceptual space.

3.1. Acquiring data about place types

The set E_{place} of place types that we have considered are those from the following place type taxonomies:

GeoNames [9] organises 667 place types in 9 categories, encompassing both man-made and natural features.

Foursquare [10] also uses 9 top-level categories, but focuses mainly on urban man-made places such as restaurants, bars and shops. Although a few of these categories include sub-categories, the taxonomy is mostly flat, and we will only consider the top-level categories in this paper. In total, the Foursquare taxonomy contains 435 place types.

OpenCYC [11] is a common-sense knowledge base, containing a large open-domain taxonomy. To derive a suitable place type taxonomy from OpenCYC, we considered all refinements of the category *Site*, leading to a total of 3388 place types, organised in directed acyclic graph.

We have used Flickr[12], a photo-sharing website, to derive textual information about each of the place types. Users on Flickr can assign tags (i.e. short textual descriptions) to their photos. Our assumption is that photos which are tagged with a given place type (e.g. *restaurant*) will often contain other tags that relate to that place type (e.g. *food*, *waiter*, *dessert*). The distribution of tags that co-occur with a given place type on Flickr may thus provide us with meaningful information about the meaning of that place type.

We have used the Flickr API to collect a large number of photos which are tagged with one of the considered place types. For composite names such as "football stadium", photos with both of the tags *football* and *stadium* were accepted, in addition to those including the concatenation of the whole name,

[9] `http://www.geonames.org/export/codes.html`, accessed September 2013.

[10] `http://aboutfoursquare.com/foursquare-categories/`, accessed September 2013.

[11] `http://www.cyc.com/platform/opencyc`, accessed April 2014.

[12] `https://www.flickr.com`

footballstadium. In total we collected 22 816 139 photos in April 2014. Place types with fewer than 1000 associated photos on Flickr have been removed from the set E_{place} of considered entities. Sufficient numbers of photos were found for 391 place types from Foursquare, 403 place types from GeoNames, and 923 place types from OpenCYC. For each of the remaining entities e, we define the associated text document D_e as the bag-of-words containing all tags that co-occur with e in the collected sample of Flickr photos.

3.2. Acquiring data about movies

Initially, we considered the 50 000 movies with the highest number of votes on IMDB[13]. For each of these movies, in October 2013 we collected reviews from the following sources: IMDB[14], Rotten Tomatoes[15], SNAP project's Amazon reviews [50][16], and the data set from [51][17]. To link reviews across these sources, we assume that movies with the same title and release year are identical; Rotten Tomatoes was linked with IMDB by using the IMDB IDs that are provided by the Rotten Tomatoes API. We then selected the 15 000 movies whose associated reviews contained the highest number of words as the set E_{movie} of considered movies. Similarly as in Section 3.1, we associated with each movie e from E_{movie} a bag-of-words D_e, now consisting of the terms from the reviews rather than tags of associated Flickr photos. We have removed words from a standard list of stop words[18], converted all words to lower case, and removed diacritics and punctuation.

3.3. Acquiring data about wines

For wines, we used the corpus of wine reviews from the SNAP Project[19]. This corpus contains 2 025 995 reviews of 485 179 different wines. For each of these wines, the name of the corresponding wine variant is also provided, e.g. the wine *2001 Thierry Allemand Cornas Reynard* is of variant *Syrah*. The entities we will consider in this paper are these wine variants, rather than

[13] According to `ftp://ftp.fu-berlin.de/pub/misc/movies/database/ratings.list.gz`

[14] `http://www.imdb.com/`reviews

[15] `http://www.rottentomatoes.com`

[16] `https://snap.stanford.edu/data/web-Amazon.html`

[17] `http://ai.stanford.edu/~amaas/data/sentiment/`

[18] `http://snowball.tartarus.org/algorithms/english/stop.txt`

[19] `https://snap.stanford.edu/data/web-CellarTracker.html`

the specific wines (since too little information is available about most of the specific wines). In particular, the set E_{wine} contains the 330 wine varieties for which the available reviews together contained at least 1000 words. The bag-of-words representation D_e for each of these wine varieties e was obtained as for the movie data.

3.4. Dimensionality reduction

The process explained in Sections 3.1–3.3 results in a set of entities E from a given domain and for each entity $e \in E$ a document D_e, represented as a bag of words. To obtain a vector-space representation of these documents, we need to quantify for each term occurring in the corpus $\{D_e \,|\, e \in E\}$ how strongly it is associated with e. Following [52], we use the Positive Pointwise Mutual Information (PPMI) measure to this end. In particular, let $c(e,t)$ be the number of times term t occurs in the document D_e. Then the weight $ppmi(e,t)$ for term t in the vector representing e is given by $\max\left(0, \log\left(\frac{p_{et}}{p_{e*} \cdot p_{*t}}\right)\right)$ where

$$p_{et} = \frac{c(e,t)}{\sum_{e'} \sum_{t'} c(e',t')} \qquad p_{e*} = \sum_{t'} p_{et'} \qquad p_{*t} = \sum_{e'} p_{e't}$$

Like the popular TF-IDF measure, PPMI will favor terms which are frequently associated with the entity e while being relatively infrequent in the corpus overall. Let us use $\mathbf{v}_e$ to denote the resulting vector representation of entity e, i.e. if the considered terms are $t_1, ..., t_k$ then $\mathbf{v}_e = (ppmi(e,t_1), ..., ppmi(e,t_k))$. There are two reasons why we cannot use these vector representations directly. First, these representations are too sparse: often $ppmi(e,t) = 0$ will hold even if t is relevant to the entity e, because t has not been mentioned in the document D_e. Second, as will become clear in Section 4, we need a geometric representation in which entities correspond to points and in which Euclidean distance is a meaningful measure of dissimilarity (which implies that spatial relations such as betweenness and parallelism are also meaningful).

To address both issues we use multi-dimensional scaling (MDS). MDS takes as input a dissimilarity matrix and a number of dimensions n. To measure the dissimilarity between two entities e_i and e_j we use the normalized angular difference:

$$ang(e_i, e_j) = \frac{2}{\pi} \cdot \arccos\left(\frac{\mathbf{v}_{e_i} \cdot \mathbf{v}_{e_j}}{\|\mathbf{v}_{e_i}\| \cdot \|\mathbf{v}_{e_j}\|}\right)$$

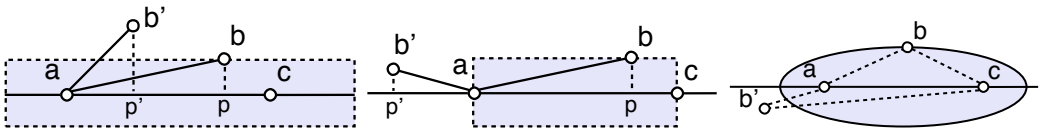
Given these dissimilarities, MDS generates an n-dimensional Euclidean space, in which each entity e_i is associated with a point p_i such that the Euclidean distance $d(p_i, p_j)$ approximates the dissimilarity $ang(e_i, e_j)$. We will consider Euclidean spaces of dimensions 20, 50, 100 and 200. In general, using a small value for n leads to representations which mainly capture high-level properties of the entities, and thus a better generalization of specific representations. On the other hand, by using a larger value of n, the representations preserve more specific details, at the cost of being more noisy. The number of dimensions n thus reflects a trade-off. We have used the implementation of classical multidimensional scaling from the MDSJ java library[20].

Several authors have already proposed the use of dimensionality reduction for commonsense reasoning. For example, [4] uses Singular Value Decomposition (SVD) to find missing properties in ConceptNet. However, SVD produces a representation in which entities correspond to vectors, which should be compared in terms of cosine similarity rather than Euclidean distance. In applications which only rely on similarity, this poses no problems. However, we can expect that spatial relations such as betweenness and parallelism (which we will need) are not meaningful in the representations derived from SVD. This has been confirmed by experiments in [15], where we compared the representations resulting from MDS, SVD, and Isomap [53]. While it may be possible to find alternative measures of betweenness and parallelism that make sense for vectors, the use of point representations is more intuitive and will allow us to use off-the-shelf SVM classifiers for identifying salient directions in Section 4.2.2, among others.

In the following, we will refer to $\mathcal{S}_{place}$, $\mathcal{S}_{movie}$ and $\mathcal{S}_{wine}$ to denote the Euclidean spaces that were obtained using this process (assuming the number of dimensions is clear from the context, or irrelevant for the discussion). We will refer to these spaces as conceptual spaces. Sometimes we will write $\mathcal{S}$ to denote a generic conceptual space. The point in $\mathcal{S}$ corresponding to an entity e will be denoted by p_e. However, when there is no cause for confusion, we will often use the notation e to refer both to the entity and the corresponding point in $\mathcal{S}$. We have made all conceptual space representations, as well as the initial PPMI weighted vectors, available online[21].

[20] `http://www.inf.uni-konstanz.de/algo/software/mdsj/`

[21] `http://www.cs.cf.ac.uk/semanticspaces/`



(a) *Col* (b) Btw_A (c) Btw_B

Figure 2: Comparison of the collinearity and betweenness measures. The shaded area depicts all points b that are between a and c at least to a particular degree. The point b' is considered to be between a and c to a lesser degree than b.

4. Deriving semantic relations from conceptual spaces

It is well-known that Euclidean distance in $\mathcal{S}$ can be used to define a measure of semantic similarity, e.g. $sim(a, b) = e^{-\lambda \cdot d(a,b)}$ for $a, b \in E$ and some fixed $\lambda > 0$. This particular similarity measure is often used in the field of cognition and has proven useful in models of human categorization [54, 55]. Beyond similarity, few authors have looked at modelling semantic relatedness using spatial relations (apart from the preliminary work in [44]). To characterise additional forms of semantic relatedness, we will consider spatial relations in $\mathcal{S}$ that relate to betweenness and direction, which will allow us to implement classifiers based on interpolative and a fortiori reasoning.

4.1. Betweenness

We say that an entity b is conceptually between entities a and c if b has all the *natural* properties that a and c have in common. Geometrically, we say that the point p_b is between points p_a and p_c if $\cos(\overrightarrow{p_a p_b}, \overrightarrow{p_b p_c}) = 1$, where we write $\cos(\mathbf{x}, \mathbf{y})$ for vectors $\mathbf{x}$ and $\mathbf{y}$ to denote the cosine of the angle between $\mathbf{x}$ and $\mathbf{y}$. Both notions of betweenness can be linked to each other, by considering that natural properties tend to correspond to convex regions in conceptual spaces [9]. Indeed, p_b is geometrically between p_a and p_c iff all convex regions that contain p_a and p_c also contain p_b. This suggests that we can identify geometric betweenness in $\mathcal{S}$ with conceptual betweenness.

Since it is unlikely that a point b will be perfectly between two other points a and c in $\mathcal{S}$, we need to measure degrees of betweenness. First we define a degree of collinearity as follows:

$$Col(a, b, c) = \|\overrightarrow{bp}\|$$

where p is the orthogonal projection of b on the line connecting a and c, as illustrated in Figure 2(a). To measure betweenness, we additionally check

Table 1: Examples of places b that were found to be (largely) between two other places a and c (using measure Btw_A).

Place type b	Places (a, c)	Place type b	Places (a, c)
american rest.	(fast food rest., french rest.)	abbey	(castle, chapel)
bistro	(restaurant space, tea room)	bog	(heath, wetland)
butcher shop	(marketplace, slaughterhouse)	bookstore	(mall, newsstand)
cafe	(coffee shop, restaurant)	conservatory	(greenhouse, playhouse)
deli	(bakery, fast food restaurant)	duplex	(detached house, triplex)
wine shop	(gourmet shop, liquor store)	garden	(flowerbed, park)
furniture store	(home store, vintage store)	gift shop	(flower shop, toy store)
grocery store	(convenience store, farmers market)	manor	(castle, mansion house)
science museum	(history museum, planetarium)	rice paddy	(bamboo forest, cropland)
sushi rest.	(japanese rest., tapas rest.)	flower shop	(garden center, gift shop)

whether p is between a and c:

$$Btw_A(a, b, c) = \begin{cases} Col(a, b, c) & \text{if } \cos(\overrightarrow{ac}, \overrightarrow{ab}) \geq 0 \text{ and } \cos(\overrightarrow{ca}, \overrightarrow{cb}) \geq 0 \\ +\infty & \text{otherwise} \end{cases}$$

noting that $\cos(\overrightarrow{ac}, \overrightarrow{ab}) \geq 0$ and $\cos(\overrightarrow{ca}, \overrightarrow{cb}) \geq 0$ iff p lies on the line segment between a and c. If $Btw_A(a, b, c) = 0$ then b is perfectly between a and c, with higher scores corresponding to weaker betweenness relations. The measures Col and Btw_A are illustrated in Figures 2(a) and 2(b).

We also consider a second betweenness measure, based on the observation that $\|\overrightarrow{ac}\| \leq \|\overrightarrow{ab}\| + \|\overrightarrow{bc}\|$ (by the triangle inequality), and $\|\overrightarrow{ac}\| = \|\overrightarrow{ab}\| + \|\overrightarrow{bc}\|$ iff b is exactly between a and c:

$$Btw_B(a, b, c) = \frac{\|\overrightarrow{ab}\|}{\|\overrightarrow{ab}\| + \|\overrightarrow{bc}\|}$$

In contrast to Btw_A, higher values for Btw_B represent a stronger betweenness relation, with a score of 1 denoting perfect betweenness. With this alternative definition, illustrated in Figure 2(c), points near a or b will get some degree of betweenness, even if their projection p is not between a and b.

Table 1 shows for a number of place types b which pair of place types (a, c) minimizes $Btw_A(a, b, c)$. Most of these triples indeed intuitively correspond to conceptual betweenness. For example, properties which hold for convenience stores and farmers markets (e.g. selling fruit) also tend to hold for grocery stores. In addition to these examples, we have also found several triples (a, b, c) with a low score for $Btw_A(a, b, c)$ where a, b and c are highly similar,

but none of the place types is clearly between the other two (not shown in the table). Examples include:

(*tanning salon*, *nail salon*, *yoga studio*)

(*ski chairlift*, *ski lodge*, *ski trail*)

(*chinese restaurant*, *malaysian restaurant*, *indonesian restaurant*)

It is indeed easy to see that if a, b and c are close to each other in $\mathcal{S}$, at least one of $Btw_A(a, b, c)$, $Btw_A(b, a, c)$ and $Btw_A(a, c, b)$ will be low. While such triples do not always correspond to our intuition of betweenness, they tend to be useful for implementing interpolative reasoning, as we will see in Section 6. We also found some triples (a, b, c) in which one of the place types is a generalization of the other(s). Examples include:

(*toy store*, *game store*, *video game store*)

(*thai restaurant*, *restaurant*, *vietnamese restaurant*)

(*rainforest*, *temperate rainforest*, *temperate forest*)

To more accurately model the relationship between place types at different levels of granularity, we could represent place types as regions, instead of points, and identify approximate betweenness, overlap, and part-of relations between these regions. Such representations can be obtained by clustering the text documents associated with each entity, representing each such cluster as a point in $\mathcal{S}$ and then identifying the entity e.g. with the convex hull of these points (possibly after removing outliers). Initial experiments, reported in [15], have revealed that such a region based representation did not consistently improve the performance of a betweenness based classifier, while being computationally much more expensive. Therefore, we have not considered region based representations in this paper. An alternative would be to define betweenness relative to a taxonomy of place types, and only consider triples between place types that are at the same level of the taxonomy.

Tables 2 and 3 provide examples of betweenness for movies and wine varieties. The examples for movies closely correspond to an intuitive notion of conceptual betweenness. For wines, however, we mainly seem to find triples (a, b, c) where b is either highly similar to a or to c.

4.2. Interpretable directions

It is tempting to think of the dimensions of the space $\mathcal{S}$ as primitive features from the domain of interest. Unfortunately, however, the dimensions

Table 2: Examples of movies b that were found to be (largely) between two other movies a and c (using measure Btw_A); only 300 most popular movies on IMDB have been considered.

Movie b	Movies (a, c)
aliens	(star trek, cloverfield)
blade runner	(the wizard of oz, 2001: a space odyssey)
cast away	(titanic, into the wild)
edward scissorhands	(beauty and the beast, forrest gump)
forest gump	(million dollar baby, stand by me)
good will hunting	(dead poets society, rain man)
lord of the rings: the fellowship of the ring	(harry potter and the prisoner of azkaban, troy)
mission impossible	(the rock, skyfall)
scarface	(sin city, the godfather)
shrek 2	(wedding crashers, the lion king)
star wars: episode vi - return of the jedi	(the lord of the rings: the two towers, star trek)
troy	(braveheart, thor)
unbreakable	(sin city, the sixth sense)
wall-e	(monsters inc., 2001: a space odyssey)

Table 3: Examples of wines b that were found to be (largely) between two other wines a and c (using measure Btw_A).

Wine b	Wines (a, c)	Wine b	Wines (a, c)
barbaresco	(barbera, barolo)	barbaresco	(barolo, dolcetto)
cabernet sauvignon	(merlot, zinfandel)	chablis	(montrachet, muscadet)
chenin blanc	(sancerre, vouvray)	merlot	(cabernet sauvignon, malbec)
montepulciano	(chianti, pinotage)	montrachet	(chablis, meursault)
petite sirah	(petit verdot, zinfandel)	pinot gris	(gewurztraminer, pinot blanc)
riesling	(gewurztraminer, spatlese)	vacqueyras	(cahors, gigondas)
vouvray	(chenin blanc, muscadet)	zinfandel	(merlot, petite sirah)

that we obtain from MDS tend not to have an intuitive meaning. Most existing work on learning spaces with interpretable dimensions has focused on non-negative matrix factorization (NMF [56]). The advantage of NMF stems from the fact that each dimension in the learned space corresponds to a linear combination of features from the original space (i.e. natural language terms in our context) which uses positive weights only. The positive nature of the weights means that dimensions can be seen as (weighted) clusters of terms. Moreover, some approaches to NMF explicitly enforce sparsity to obtain dimensions which correspond to linear combinations of just a few terms, and thus further improve the interpretability of the learned dimensions [57]. As is the case for SVD, however, Euclidean distance is not necessarily meaningful as a measure of dissimilarity in the space obtained by NMF.

Since Euclidean distance plays a key role in our approach (e.g. given its relationship to the notion of betweenness), instead of using NMF, we will show how we can identify (in an unsupervised) way interpretable directions

in the space $\mathcal{S}$, corresponding to the most salient properties of the domain of interest (but typically not orthogonal to each other). For example, in a space of movies there could be a direction pointing towards *more violent* movies. Each of the identified directions will yield a ranking of the considered entities, according to how much they have the corresponding property, e.g. by identifying the direction modelling 'more violent than' in a conceptual space of movies, we can obtain a ranking of movies according to their level of violence. These rankings provide a purely qualitative representation of the conceptual space $\mathcal{S}$, capturing semantic relations which are not yet included in existing knowledge bases such as Freebase and YAGO.

In Section 4.2.1 we explain how we can identify directions that correspond to interpretable properties, while in Section 4.2.2 we discuss how these directions can be used to make explicit how one entity is semantically related to another. Section 4.2.3 then focuses on selecting those directions that correspond to the most salient properties.

4.2.1. Interpreting semantic relations as directions

Interpretable directions should correspond to natural language terms. It is moreover natural to assume that such directions will correspond to terms that occur in the text corpus from which the space $\mathcal{S}$ was induced[22]. As a first step, we therefore compile a set T of all terms that are sufficiently frequent in this text corpus. Let T_{place}, T_{movie} and T_{wine} (defined as follows) be the set of terms that are considered for $\mathcal{S}_{place}$, $\mathcal{S}_{movie}$ and $\mathcal{S}_{wine}$ respectively.

In the case of $\mathcal{S}_{place}$, where the text corpus consists of Flickr tags, T_{place} was chosen as the set of all tags that co-occur with at least 50 different place types in our Flickr corpus (out of the 1383 considered place types). This resulted in a total of $|T_{place}| = 21\,833$ candidate terms. In the case of $\mathcal{S}_{movie}$, we considered adjectives, nouns, adjective phrases and noun phrases that appear in the corpus of reviews. The underlying assumption is that there will be meaningful directions of two types. Some dimensions will correspond to gradual properties (e.g. violent, funny, creepy), which are most likely to correspond to an adjective or adjective phrase. Other dimensions will correspond to topics, which may relate to the genre, theme or other aspects of the movie that are likely to correspond to a noun or noun phrase. To

[22]For some abstract properties, the most appropriate term may not occur in the corpus. In such cases, external resources such as WordNet or Wikipedia could be used to identify additional terms that are relevant for the considered domain.

select adjectives, nouns, and adjective/noun phrases from the reviews, we used the part-of-speech tagger and chunker from the Open NLP Project[23]. We only considered words and phrases which appear in the reviews of at least 100 movies (out of the 15 000 considered movies), resulting in a total of 22 903 candidate terms. The thresholds of 50 in the case of T_{place} and 100 in the case of T_{movie} were chosen such that the total number of candidate terms is approximately equal, i.e. $T_{place} \approx T_{movie}$. Finally, in the case of $\mathcal{S}_{wine}$ we again extracted adjectives, nouns and adjective/noun phrases from the reviews and considered all terms which appear in the reviews of at least 50 different wine varieties (out of the 330 considered varieties), leading to $|T_{wine}| = 6385$ candidate terms.

We then assign a direction in $\mathcal{S}$ to each term t from T. To this end, we first train a (linear) SVM to find the hyperplane H_t in $\mathcal{S}$ that best separates the entities to which t applies from the others, where we say that t applies to an entity if at least one of its associated text documents contains t. The perpendicular vector $\overrightarrow{v_t}$ of H_t is then considered to define the direction associated with t. We used LibSVM[24] with a linear kernel and the standard values for all parameters, but we adapted the costs of the training instances to deal with class imbalance (using the ratio between entities with/without the term as cost). Only some of the terms in T correspond to properties of the considered entities. Accordingly only some of the terms in T can be faithfully modelled as a direction in $\mathcal{S}$. Therefore, as a last step, we estimate to what extent $\overrightarrow{v_t}$ is indeed a meaningful representation of the term t. Here we use the assumption that the better H_t separates entities to which t applies from the others in $\mathcal{S}$, the better $\overrightarrow{v_t}$ models the term t. To quantify the performance of the SVM model, we used Cohen's Kappa measure [58], due to its tolerance to class imbalance. We also considered several alternative metrics, including Spearman's and Kendall's correlation coefficients to measure the correlation between the ranking induced by $\overrightarrow{v_t}$ and the number of times t appears in the documents associated with each entity. As the results from these alternative metrics were less promising in initial experiments, we have not considered them further. The higher the Kappa score of a term t, the more we consider $\overrightarrow{v_t}$ to be a faithful representation of the term t. We write T^λ for the set of terms from T whose Kappa measure is at least λ.

[23] `http://opennlp.apache.org/`

[24] `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`

4.2.2. Describing the semantic relation between two entities

For two entities e_1 and e_2, represented as points in $\mathcal{S}$, we can compare the vector $\overrightarrow{e_1e_2}$ with the vectors $\overrightarrow{v_t}$ of interpretable terms t ($t \in T^\lambda$ for a given value of $\lambda > 0$). This leads to the following measure diff_A of how well term t describes how entity e_2 differs from entity e_1.

$$\mathit{diff}_A(e_1, e_2; t) = \cos(\overrightarrow{e_1e_2}, \overrightarrow{v_t})$$

As a baseline method, we can also look at how frequently t is used in the text documents associated with e_1 and e_2. As before, let $\mathit{ppmi}(e, t)$ be the PPMI-value of term t in the document associated with entity e. We define:

$$\mathit{diff}_B(e_1, e_2; t) = \mathit{ppmi}(e_2, t) - \mathit{ppmi}(e_1, t)$$

Note that diff_B is only defined for unigrams, in contrast to diff_A.

Tables 4-6 illustrate the kind of results that both measures can achieve (using $\lambda = 0.1$ and a 100-dimensional space), while a more formal evaluation based on a crowdsourcing experiment will be presented in Section 6.2.1. The examples in Table 4 show that diff_A can indeed find meaningful labels to describe how two place types differ. As the example for *cupcake shop* illustrates, the terms which are selected also depend on the choice of e_1: a *cupcake shop* mainly differs from a *hobby shop* in the fact that it sells confections, leading to terms such as *icing* and *dessert*. On the other hand, *cupcake shop* mainly differs from a *bagel shop* in the kind of confections that are sold, leading rather to terms such as *gift* and *handmade*. The baseline measure diff_B tends to select terms which are too specific (e.g. *motoq9c*), although some highly relevant terms are identified as well (e.g. *caribbeanfood*, *pescatarian*, *medicaleducation*). In Table 5, the use of phrases has the advantage that terms such as *science fiction* can be recognized. On the other hand, it also leads to the occurrence of phrases such as *your spine* (from the idiom "sending shivers down your spine") which are less suitable as descriptions. The results for wines in Table 6 are mixed. While diff_A is able to identify reasonable terms when the two wines are very different (e.g. *chardonnay* and *merlot*), for wines that are more similar (e.g. *pinot gris* and *pinot blanc*), no terms are found which achieve a high value for $\mathit{diff}_A(e_1, e_2; t)$, resulting in some generic terms being identified among the top terms (e.g. *a tasting note*). This may be explained by the fact that the wine space contains only 330 entities (making the problem of inducing a 100-dimensional conceptual space under-constrained), the fact that T_{wine} contains fewer candidate terms

than T_{place} and T_{movie} (potentially leading to fewer interpretable directions in the space), and the fact that we have less text, on average, per wine than per film of place type. Moreover, wines from the same variety can be very different, which further complicates the problem of deriving meaningful conceptual space representations.

4.2.3. Selecting the most salient directions

Much of human reasoning relies (only) on our ability to rank entities or concepts according to a particular feature [59]. A central problem is thus to identify the most salient properties of a given domain. In our setting, this boils down to selecting the most salient directions in $\mathcal{S}$. We can intuitively think of these directions as a non-orthogonal basis for the space $\mathcal{S}$, where dimensions now correspond interpretable properties.

To select interpretable directions, we will only consider the terms in $T^{0.5}$, as a Kappa score of 0.5 was found in initial experiments to offer a good balance between keeping a sufficient number of terms and ensuring that the terms are modelled adequately as directions in $\mathcal{S}$. To select the most salient directions, we first select the term with the highest Kappa score overall. Then we repeatedly apply the following process: as the i^{th} term, we select the term t minimising $\max_{j<i} \cos(\overrightarrow{v_{t_j}}, \overrightarrow{v_t})$. In other words, we repeatedly select the term which is least similar to the terms that have already been selected. One possibility would be to choose as many terms as there are dimensions in $\mathcal{S}$. However, because we have no guarantees that all terms will be linearly independent from the others, we allow for some redundancy, and select $2n$ terms for an n-dimensional space.

Let $d_1, ..., d_{2n}$ be the terms that have been selected. We then associate with each term d_i a cluster C_i containing all terms from $T^{0.1}$ which are more similar to d_i than to any of the other directions d_j. The cluster C_i in turn defines a direction $\overrightarrow{v_i^*} = \frac{1}{|C_i|} \sum_{t \in C_i} \overrightarrow{v_t}$, which we will consider as the direction corresponding to term d_i. Note that in principle, we could now reassign the terms in $T^{0.1}$ to the closest direction v_i^*, as in the k-means algorithm, but initial experiments with this approach did not yield any clear improvements.

Each of these salient directions $\overrightarrow{v_i^*}$ naturally induces a ranking. In particular, for a given C_i, let $L_i = \{o + \lambda \cdot \overrightarrow{v_i^*} \mid \lambda \in \mathbb{R}\}$ be the corresponding line, where o represents the origin of $\mathcal{S}$. For each entity $e \in E$, let q_e^i be the orthogonal projection of p_e on the line L_i. Then q_e^i is of the form $q_e^i = o + \lambda_e^i \cdot \overrightarrow{v_i^*}$. The associated ranking $<_i$ is defined as follows: $e <_i f$ iff $\lambda_e^i < \lambda_f^i$. We write $r_i(e)$

Table 4: Top terms identified by $diff_A$ and $diff_B$ for pairs of place types.

Place e_2	Place e_1	$diff_A$	$diff_B$
american restaurant	scandinavian restaurant	diner, fries, deli, burger	foodfashion, pakistaniamerican, olypa, twincitiesrestaurantblog
antique shop	bike shop	homedecor, dollhouse, shelf, fabric	lucketsstoreantiques, petitsdetails, framedphotos, candypopimages
asian restaurant	scandinavian restaurant	noodles, prawn, chopsticks, noodle	badmenu, funnymenu, wrongmenu, weirdfood
caribbean restaurant	brazilian restaurant	carribean, tropics, bahamas, whitesand	frenchside, mambogrill, caribbeanfood, laguarida
cupcake shop	bagel shop	gift, handmade, etsy, cookie	cupcakesqatar, sugarbabys, cupcakesdoha, flashcolouricing
cupcake shop	hobby shop	icing, dessert, dough, chocolate	cupcakesdoha, cupcakesqatar, flashcolouricing, crumbs
elementary school	law school	elementary, childhood, playground, kid	vsb, 5thgrade, carlynton, yeongju
home bar	cocktail bar	home, apartment, bedroom, house	motoq9c, homebarista, q9c, retrobar
medical school	law school	illness, doctor, disease, doctors	medicaleducation, medschool, schoolofmedicine, balliolmedicalsociety
restaurant bar	italian restaurant	happyhour, bartender, whisky, guinness	bedsupperclub, 1s22s22p6, burningmen, bangkokinvaders
seafood restaurant	indian restaurant	schrimp, prawn, lobster, tuna	pescatarian, crabhouse, mccormickschmicks, southeastasiafoodtrip
sports bar	cocktail bar	sports, stadium, basketball, athletics	vgolf, trepp, fujixerox, documentsolutionsgroup

Table 5: Top terms identified by $diff_A$ and $diff_B$ for pairs of movies.

Movie e_2	Movie e_1	$diff_A$	$diff_B$
alien	american history x	a spaceship, science fiction, the technology, the special effects	nostromo, chestburner, gigers, facehugger
alien	star trek	a horror film, a horror movie, your spine, horror	nostromo, chestburner, gigers, facehugger
american beauty	good will hunting	suburbia, adultery, a black comedy, infidelity	burnhams, fitts, thora, lesters
django unchained	black swan	shootouts, gunplay, a summer blockbuster, race relations	tarantinos, spaghetti, slavery, christoph
fight club	gladiator	conformity, society, voyeurism, a dark comedy	durden, uhls, testicular, ikea
pulp fiction	inception	organized crime, absolutely hilarious, pretty funny, a great collection	marsellus, winnfield, slims, marcellus
pulp fiction	up	graphic violence, the gangs, the gangsters, the squeamish	marsellus, winnfield, slims, marcellus
jurassic park	kill bill: vol. 1	dinosaurs, an expedition, the scientist, these creatures	nedry, sattler, paleobotanist, ariana
requiem for a dream	die hard	drug addiction, drugs, addiction, a downward spiral	goldfarb, selby, tibbons, burstyns
schindler's list	american beauty	atrocities, the nazis, concentration camps, genocide	goeth, plaszow, itzhak, amon
shutter island	goodfellas	psychological thrillers, the mystery, spooky, psychological horror	cawley, aule, solando, ashecliff
sin city	avatar	repulsive, disgusting, his best film, pulp	hartigan, marvs, shellie, miho
sin city	reservoir dogs	the comics, comic books, the visual effects, novels	hartigan, marvs, shellie, miho
v for vendetta	american history x	political intrigue, espionage, the battle sequences, socialism	fawkes, stutler, mcteigue, wachowski

Table 6: Top terms identified by $diff_A$ and $diff_B$ for pairs of wines.

Place e_2	Place e_1	$diff_A$	$diff_B$
barbera	barolo	every day wine, acid medium, short length, a good quaffer	quorum, coppo, scarrone, braida
barolo	barbera	a remarkable wine, sweet taste, half way, some melon	brunate, pira, borgogno, rionda
barolo	beaujolais	the very long finish, noticeable acidity, at least 5 more years, the ability	brunate, cannubi, monfortino, rocche
cabernet sauvignon	beaujolais	the very long finish, sweet taste, vanilla notes, the steak	merus, butterdragon, dunn, corison
chardonnay	merlot	light straw color, yellow apples, pale yellow color, golden yellow color	chards, auberts, ritchie, fuisse
chardonnay	muscat	very mineral, sea breeze, green fruit, good fruit and acidity	chards, ritchie, fuisse, kistlers
chardonnay	pinot blanc	my husband, a curiosity, a cellar, minimal notes	chards, auberts, ritchie, melolactic
chardonnay	pinot gris	sweet oak, great pleasure, 60 minutes, earthy minerality	chards, auberts, ritchie, kistlers
pinot gris	pinot blanc	really sure, low expectations, a tasting note, the best bottles	grigios, sgns, pgs, rubarbe, sgn
tempranillo	beaujolais	very new world, a long decant, first impression, first glass	rdd, pesquera, duero, aalto

Table 7: Examples of salient directions found in the conceptual space of place types. For each salient direction, we also list some of the terms in the corresponding cluster.

nature	glacial, geology, wilderness, forests, animal, naturalworld, peaceful ...
chicago	denver, queens, texas, newyork, boston, vancouver, uptown, sf, ...
aircraft	flughafen, 737, controltower, boeing, landing, ...
sauna	massage, chalet, wellness, piscine, hottub, jaccuzzi, luxuryhotel, ...
coal	electricity, railways, steel, furnace, industry, pollution, ...
bike	motorcycle, cyclist, scooter, ducati, lane, busstop, ...
lettuce	sandwich, tuna, noodles, poultry, tomatoes, lamb, sausage, steak, ...
sailingboat	fjord, motorboat, lakefront, windsurfing, sunshinecoast, mooring, ...
finance	jobs, investment, canarywharf, officebuilding, skyscraper, ...
science	learning, biology, classroom, laboratory, physics, nasa, planetarium, ...
malaysia	vietnamese, southkorea, guangzhou, philippines, sydney, backpacker, ...
light	photography, wide, perspective, digital, lines, photoshop, ...
uk	victorian, southyorkshire, cardiff, unitedkingdom, abderdeen, somerset, ...
illness	therapy, nurse, medicine, doctors, clinic, healthcare, stress, ...
bahnhof	eurostar, intercity, station, busstation, londonunderground, trainstation, ...
cave	rock, quarry, abyss, chasm, limestone, rockformation, stalactite, ...
pub	publichouse, inn, bar, ale, tavern, ...
room	office, chair, living, window, furniture, kitchen, hotel, bedroom, ...
abandoned	derelict, dilapidated, vacant, ghosttown, disused, graffiti, rusty, creepy, ...
mallard	parkbench, squirrel, aligator, everglades, goose, trout, citypark, swans, ...
tapas	barceloneta, ristorante, margarita, cerveza, olives, piraeus, alfresco, ...

for the rank of entity e in this ranking, i.e. $r_i(e) = |\{f \mid f \in E, f <_i e\}|$. The complete list of clusters and the corresponding directions, for each of the conceptual spaces has been made available on the online companion website[25]. Moreover, we have also made available for each entity e the feature vector $(\lambda_e^1, ..., \lambda_e^{2n})$ (from which the rankings $<_i$ can readily be obtained). Examples of the selected salient directions and the correspond clusters are show in Table 7 for place types, Table 8 for movies and Table 9 for wines. Many of the directions in Table 7 encode properties of place types. Examples include *nature*, *bahnhof*, *pub* and *tapas*. Other clusters correspond to geographic areas, e.g. *chicago*, *malaysia* and *uk*. Given that $\mathcal{S}_{place}$ has been derived from Flickr, it is not surprising to see some clusters related to photography, such as *light*. The directions about movies in Table 8 are different in a number of aspects. Being derived from full text documents instead of tags, the table contains individual terms as well as phrases. More importantly, the fact that adjectives

[25] `http://www.cs.cf.ac.uk/semanticspaces/`

are now also considered seems to lead to a higher number of directions which describe properties of movies, e.g. *touching*, *clever*, *romantic*, *eerie*, etc. In addition, we also find directions corresponding to movie themes, e.g. *horror movies*, *supernatural* or *political*. Some of the other directions include *budget* (referring to the production value), directions grouping phrases that start with *her* and *his* (referring to whether the lead actor is male or female), *vhs* (referring to older films, which were initially released on e.g. VHS or LaserDisc), *era* (referring to films which are set in the past) and *sequel* (referring to films which are part of a series). The directions in Table 9 mainly correspond to different flavours found in wine, which is unsurprising since reviews from which the conceptual space $\mathcal{S}_{wine}$ was induced are essentially tasting notes. In contrast to the case of place types and movies, for wines we find several clusters which are quite similar. For example, the cluster for *light food* mentions *salad* whereas the cluster for *grass* mentions *salads*. Conversely, some clusters should ideally be split into two or more separate clusters (e.g. in the cluster *grape juice*, the terms *terrible* and *a very good price* correspond to different properties). Most importantly, some natural properties of wines are lacking. Ideally there would be directions ordering wines according to their amount of tannins, the heaviness of their body, and their acidity. None of the identified directions exactly models these properties, although some directions are highly correlated with them (e.g. *dark fruits* for the amount of tannins).

4.3. Parallelism

As we discussed in Section 4.2, the vector $\overrightarrow{ab}$ defined by two entities a and b encodes how these entities differ from each other. Given four entities a, b, c and d, we can thus naturally model the relational similarity between (a, b) and (c, d) by comparing the vectors $\overrightarrow{ab}$ and $\overrightarrow{cd}$. In [29] the following measure of analogical dissimilarity is proposed:

$$diss_A(a, b, c, d) = \|\overrightarrow{cd} - \overrightarrow{ab}\| = \|(a + d) - (b + c)\| \tag{1}$$

This measure evaluates to 0 if the points a, b, c, d define a parallelogram. We can think of the direction of $\overrightarrow{ab}$ as encoding in which aspects a differs from b (e.g. 'b is more violent than a') while $\|\overrightarrow{ab}\|$ measures the amount of difference (e.g. how much more violent b is). This means that (1) not only requires that (a, b) and (c, d) are related in similar ways, but also that the amount

Table 8: Examples of salient directions found in the conceptual space of movies. For each salient direction, we also list some of the terms in the corresponding cluster.

horror movies	zombie, much gore, slashers, vampires, scary monsters, ...
supernatural	a witch, ghost stories, mysticism, a demon, the afterlife, ...
scientist	experiment, the virus, radiation, the mad scientist, ...
criminal	the mafia, robbers, parole, the thieves, the mastermind, ...
the animation	the voices, drawings, the artwork, the cartoons, anime, ...
touching	inspirational, warmth, dignity, sadness, heartwarming, ...
budget	a low budget film, b movies, independent films, ...
political	socialism, idealism, terrorism, leaders, protests, equality, corruption, ...
clever	schemes, satire, smart, witty dialogue, ingenious, ...
bizarre	odd, twisted, peculiar, lunacy, surrealism, obscure, ...
predictable	forgettable, unoriginal, formulaic, implausible, contrived, ...
twists	unpredictable, betrayals, many twists and turns, deceit, ...
romantic	lovers, romance, the chemistry, kisses, true love, ...
eerie	paranoid, spooky, impending doom, dread, ominous, ...
scary	shivers, chills, creeps, frightening, the dark, goosebumps, ...
cheesy	camp, corny, tacky, laughable, a guilty pleasure, ...
she's	her apartment, her sister, her death, her family, the heroine, actress, ...
his life	his son, his quest, his guilt, a man, his voice, his fate, his anger, ...
hilarious	humorous, really funny, a very funny movie, amusing, ...
vhs	laserdisc, videotape, this dvd version, first released, this classic, ...
violence	violent, cold blood, knives, bad people, brotherhood, ...
adaptation	the stage version, the source material, the novel, ...
sequel	the trilogy, the first film, the same formula, this franchise, ...
era	the fifties, the sixties, the seventies, a period piece, the depression, ...

of change is similar. Since often only the former is relevant, we will also consider the following measure, which disregards the amount of change:

$$sim_B(a,b,c,d) = \cos(\overrightarrow{ab}, \overrightarrow{cd}) \tag{2}$$

Note that sim_B measures relational similarity, whereas $diss_A$ measures dissimilarity. In the following, we will mainly be interested in finding the points c and d in the training data that minimize $diss_A(a,b,c,d)$ or maximize $sim_B(a,b,c,d)$, given a and b. In other words, we will only use the measures $diss_A$ and sim_B to rank pairs of objects (c,d). It is easy to show that sim_B corresponds to a normalized version of $diss_A$, in the sense that $sim_B(a,b,c,d) \leq sim_B(a,b,e,f)$ iff $diss_A\left(\frac{a}{\|\overrightarrow{ab}\|}, \frac{b}{\|\overrightarrow{ab}\|}, \frac{c}{\|\overrightarrow{cd}\|}, \frac{d}{\|\overrightarrow{cd}\|}\right) \geq diss_A\left(\frac{a}{\|\overrightarrow{ab}\|}, \frac{b}{\|\overrightarrow{ab}\|}, \frac{e}{\|\overrightarrow{ef}\|}, \frac{f}{\|\overrightarrow{ef}\|}\right)$.

Note that sim_B was also used in [45] for learning analogical relations, although for vector representations instead of point based representations. In

Table 9: Examples of salient directions found in the conceptual space of wines. For each salient direction, we also list some of the terms in the corresponding cluster.

foie gras	a dessert, very sweet, a dessert, apple pie, vanilla ice cream, ...
dark fruits	very tannic, red fruits, black cherry, black fruits, blueberries, ...
very light yellow color	light gold, white fruit, citrus palate, citrus fruits, ripe pear, ...
light food	light and fruity, a salad, the short finish, sharp acidity, ...
old world style	italian food, some decanting, oak palate, moderate tannin, ...
vin	tout, bouche, sans, beaucoup, pas, est, que, plus, ...
fresh cherries	noticeable tannins, purple flowers, pepper and spice, ...
the winery	experiment, vineyard, harvest, overtones, surprisingly good, ...
grass	lemony, grassy, oil, fish, kiwi, limestone, seafood, salads, ...
light red color	red cherries, mild tannins, low tannins, light ruby, ...
lemon oil	baked apples, white flower, some butter, a chardonnay, ...
bitter almonds	oxygen, some banana, too acidic, nice acidity, marzipan, ...
strawberries	chocolate, lamb, red berries, violets, cranberries, steak, ...
nice light	floral aromatics, crisp and clean, a warm evening, spicy foods, ...
caramel	raisin, figs, brown, nuts, toffee, amber, nutmeg, orange peel, ...
the pasta	pesto, tomato sauce, veal, olive oil, fruity finish, soft acidity, ...
grape juice	not impressive, average finish, terrible, a very good price, ...
smoky	bacon, cloves, dusty, velvety, chewy, terroir, mature, decanter, ...

particular, to complete the analogical proportion $a : b :: c : x$, given a, b and c, they propose selecting the vector $\overrightarrow{v_x}$ which maximizes $\cos(\overrightarrow{v_x}, \overrightarrow{v_b} - \overrightarrow{v_a} + \overrightarrow{v_c})$ (where $\overrightarrow{v_a}$ is the vector representation of a, and similar for b, c, x). Since $\cos(\overrightarrow{v_x}, \overrightarrow{v_b} - \overrightarrow{v_a} + \overrightarrow{v_c}) = \cos(\overrightarrow{v_x} - \overrightarrow{v_c}, \overrightarrow{v_b} - \overrightarrow{v_a})$, this corresponds to applying sim_B to vector representations. There are, however, a number of differences between the aims of [45] and our aims in this paper. For example, due to the way in which the vector representations from [45] have been learned, their approach is able to recognize syntactic regularities in language, such as *better* : *best* :: *rougher* : *roughest*. They also identify semantic relations between common words, such as *man* : *woman* :: *king* : *queen*. In contrast, we focus on learning fine-grained relations between entities of the same type.

Tables 10, 11 and 12 contain examples of analogical pairs of place types, movies, and wines. In particular, each line in these tables corresponds to a tuple (a, b, c, d) for which $sim_B(a, b, c, d)$ is close to 1. Some of these tuples are such that a and c are very similar and b and d are very similar. For example, in Table 10, the tuple (*baseball diamond*, *college science building*, *stadium*, *college campus*) does not intuitively correspond to an analogy. It is found because *baseball diamond* and *stadium* are located close to each other in $\mathcal{S}_{place}$, as are *college science building* and *college campus*. As a result,

Table 10: Examples of analogical pairs of places w.r.t. sim_B.

abandoned prison	sheep fold	hospital room	veterinarian
abandoned home	scenic roadway	hospital room	overpass road
asian restaurant	italian restaurant	dim sum restaurant	salad place
bagel shop	paella restaurant	coffee shop	restaurant
bar	cafe	juice bar	dessert shop
dumpling restaurant	noodle house	deli	donut shop
dune	grassland	beach	wildlife reserve
hot spring	watercourse	botanical garden	temperate forest
language school	training camp	college library	college stadium
medical school	sanatorium	military school	military barracks

Table 11: Examples of analogical pairs of movies w.r.t. sim_B.

american beauty	american psycho	the sixth sense	saw
back to the future	back to the future part ii	the terminator	terminator 2: judgment day
blade runner	the shining	i robot	the others
life of pi	ted	inception	the hangover part ii
men in black	district 9	the fifth element	children of men
million dollar baby	requiem for a dream	rocky	trainspotting
source code	rear window	looper	psycho
the sixth sense	armageddon	the others	2012
trainspotting	snatch	requiem for a dream	pulp fiction

the vectors $\overrightarrow{ab}$ and $\overrightarrow{cd}$ are nearly identical, resulting in a high score for both $diss_A$ and sim_B. Despite not intuitively corresponding to analogies, such tuples will still be useful in an analogical classifier, as we will see further. To obtain tuples which intuitively do correspond to analogical proportions, we need to additionally require that a and c are sufficiently far apart (which also means that b and d will be far apart). An example of such a tuple is shown on the first line of Table 10: prisons are used for holding people while sheep folds are used for holding animals; hospital rooms are used for healing people, while veterinarians heal animals. Similarly, in Table Table 11, the movies *million dolar baby* and *rocky* are about boxing, while *requiem for a dream* and *trainspotting* are about drug abuse. On the other hand, *million dolar baby* and *requiem for a dream* have in common that they are much darker than *rocky* and *trainspotting*. In Table 12, for example, the tuple (*barbaresco*, *valpolicella*, *dolcetto*, *bardolino*) reflects that *barbaresco* is more tannic than *valpolicella* while *dolcetto* is typically more tannic than *bardolino*. On the other hand, *barbaresco* and *dolcetto* are both from the Piedmont region, while *valpolicella* and *bardolino* are from the Verona region.

Table 12: Examples of analogical pairs of wines w.r.t. sim_B.

barbaresco	valpolicella	dolcetto	bardolino
barbaresco	kabinett	barolo	spatlese
blaufrankisch	spatlese	zweigelt	kabinett
bourgueil	st. laurent	chinon	zweigelt
chardonnay	gruner veltliner	cabernet sauvignon	st. laurent
chardonnay	sancerre	pinot noir	gamay
chinon	spatlese	bourgueil	kabinett
dolcetto	silvaner	barbaresco	riesling
montrachet	zweigelt	meursault	blaufrankisch
silvaner	vacqueyras	riesling	grenache

5. Commonsense reasoning based classifiers

To evaluate the practical usefulness of the considered semantic relations, we will focus on their use in commonsense reasoning based classifiers, i.e. classifiers which are based on inference patterns such as interpolation and a fortiori inference. One of the main advantages of such classifiers is that we can easily generate explanations for the decisions they make.

Let $C_1, ..., C_m$ be disjoint categories and let O_i be a set of entities that are known to belong to category C_i (i.e. $O = \bigcup_i O_i$ is the available training data). We consider the problem of deciding which category is most likely to contain an unlabelled entity x. In this context, using similarity based reasoning corresponds to k-NN classification [20], i.e. we use $\mathcal{S}$ to find the k entities $y_1, ..., y_k$ from O which are most similar to x and then assign x to the category to which most of the entities y_i belong. We now discuss a number of classifiers which are based on other commonsense reasoning patterns.

5.1. Betweenness based classification

Betweenness can be used to classify objects similarly to how k-NN uses similarity. Instead of looking for entities y that are similar to x, we then look for pairs of entities (y, z) from the same category C_i such that x is approximately between y and z. The main underlying assumption is that the categories $C_1, ..., C_m$ correspond to convex regions in $\mathcal{S}$, in accordance with the theory of conceptual spaces [9]. Using this assumption, we can conclude that x belongs to the category C_i from the knowledge that (i) y and z belong to C_i and (ii) x is located between y and z. In practice, perfect betweenness is rare, which leads us to consider the pairs (y, z) which maximize the value of $Btw(y, x, z)$, where e.g. $Btw = Btw_A$ or $Btw = Btw_B$. More generally, we could also consider the top k such pairs. To classify the entity x, we then first identify the pairs $(y_1, z_1), ..., (y_k, z_k)$ that maximize $Btw(y_i, x, z_i)$ such that y_i

and z_i belong to the same category. Each of the pairs (y_i, z_i) suggests a category for the test instance. The final decision is then based by a majority vote from the k best pairs.

In the case where $k = 1$, this betweennness classifier can be generalized to a convex hull based classifier [60]. In a convex hull based classifier, every category C_i is represented geometrically as the convex hull of the points (i.e. entitites) that are known to belong to C_i. The test item x is then assigned to the category whose convex hull is closest. However, evaluating the distance between a point and a convex hull requires solving a quadratic optimization problem, which is computationally expensive in high-dimensional spaces.Moreover, while we have introduced *Col* mainly to define the measure Btw_A, as will become clear in the experiments in Section 6, using *Col* instead of Btw_A or Btw_B can also be useful in a classification setting. Such a classifier is similar in spirit to so-called affine hull based classifiers [61, 62], but is again computationally much less demanding.

5.2. Classification based on relational similarity

While several authors have proposed analogical proportion based classifiers, most work to date has focused on binary or nominal attributes. One exception is [29], where analogical dissimilarity between pairs of entities with continuous attributes is defined in terms of how close the feature vectors of the four entities are to defining a parallelogram, although no experimental results were provided on the use of this measure in a classifier. In [30] a definition of graded analogical proportion was given, based on fuzzy logic connectives. A corresponding classifier was moreover proposed, based on the idea that the more the attributes of 4 entities are in an analogical proportion, the more we can expect the class labels to be in an analogical proportion as well. However, while promising, the results that were obtained are not competitive with standard methods such as k-NN and SVM. Recently, somewhat better results have been reported in [32], although only datasets with clearly defined and relatively few attributes have been considered (the largest considered dataset has 36 attributes).

To assess to what extent such methods can be successful in our context, where entities are represented as points in a relatively high-dimensional Euclidean space, we will consider classifiers that use the measures diss_A and sim_B from Section 4.3. We will consider binary (i.e. two-class) classification problems only, even though the approach can naturally be extended to problems with a larger number of linearly ordered classes (and even more

generally, to problems where relational similarity between pairs of class labels can be measured). For $x, y, z, u \in \{0, 1\}$, analogical proportions are defined as follows [63]:

$$(x : y :: u : v) \quad \Leftrightarrow \quad ((x \to y) \equiv (u \to v)) \wedge ((y \to x) \equiv (v \to u)) \qquad (3)$$

Note that there are six tuples (x, y, u, v) in $\{0, 1\}^4$ that form an analogical proportion: $(0 : 0 :: 0 : 0)$, $(1 : 1 :: 1 : 1)$, $(0 : 0 :: 1 : 1)$, $(1 : 1 :: 0 : 0)$, $(1 : 0 :: 1 : 0)$ and $(0 : 1 :: 0 : 1)$. When the feature vectors of four objects are in an (approximate) analogical proportion, analogical classifiers consider that their class labels should also be in an analogical proportion. Let $cl(a) \in \{0, 1\}$ be the class label of object a, where 0 and 1 are interpreted as the logical constants false and true to evaluate (3). Let a be an entity whose class label is unkown. The approach from [29] then consists of the following steps:

- Find the triples (b, c, d) in the training set for which there exists a value $x \in \{0, 1\}$ that makes $x : cl(b) :: cl(c) : cl(d)$ an analogical proportion. Among these triples, find the one (b^*, c^*, d^*) that minimizes $diss_A(a, b^*, c^*, d^*)$.

- Choose the class label of a as the unique value $x \in \{0, 1\}$ that makes $x : cl(b^*) :: cl(c^*) : cl(d^*)$ an analogical proportion.

We will refer to this approach as $analog_A$. We will also consider the alternative $analog_B$ where in the first step, the triple (b^*, c^*, d^*) is chosen that minimizes $sim_B(a, b^*, c^*, d^*)$.

Note that the only two cases where the triple $(cl(b), cl(c), cl(d))$ cannot be extended to an analogical proportion are when $(cl(b), cl(c), cl(d)) = (0, 0, 1)$ and $(cl(b), cl(c), cl(d)) = (1, 1, 0)$. As we explain next, these two cases give rise to a different type of classifier, whose intuition is based on the idea of a fortiori reasoning. In particular, when $(cl(b), cl(c), cl(d)) = (0, 0, 1)$, we can think of $\overrightarrow{dc}$ as defining a direction from membership to non-membership of the considered class. Since b does not belong to the class, if $\overrightarrow{ab}$ is approximately parallel to $\overrightarrow{cd}$, we would expect that a would definitely not belong to the category, since it is obtained from b by following a direction that is associated with non-membership. Similarly, when $(cl(b), cl(c), cl(d)) = (1, 1, 0)$, the direction $\overrightarrow{dc}$ is associated with membership, hence we can expect a to belong to the category if $cl(b) = 1$ and $\overrightarrow{ab}$ is approximately parallel to $\overrightarrow{cd}$. This leads us to the following procedure:

- Find the triples (b, c, d) of entities in the training set such that $(cl(b), cl(c), cl(d)) = (0, 0, 1)$ or $(cl(b), cl(c), cl(d)) = (1, 1, 0)$. Among these triples, find the one (b^*, c^*, d^*) that maximizes $sim_B(a, b^*, c^*, d^*)$.
- If $(cl(b^*), cl(c^*), cl(d^*)) = (0, 0, 1)$, we choose $cl(x) = 0$. If $(cl(b^*), cl(c^*), cl(d^*)) = (1, 1, 0)$, we choose $cl(x) = 1$.

We will refer to this approach as $analog_C$. Since the intuition here is purely based on the direction of change, we only consider sim_B. This idea of specifically looking at types of change that affect the class label is somewhat reminiscent of the approach proposed in [64], which is based on learning change patterns between binary feature vectors that affect the class label. Note that we will refer to $analog_A$, $analog_B$ and $analog_C$ as analogical classifiers, for the ease of presentation, even though $analog_C$ is more related to a fortiori reasoning than to existing analogical proportion based classifiers.

To avoid the cubic time complexity of a naive implementation, in variants using $diss_A$, we maintain three KD trees $T^>$, $T^<$ and $T^=$ storing respectively the vectors $\overrightarrow{ab}$, with a and b entities from the training data, for which $class(a) > class(b)$, $class(a) < class(b)$ and $class(a) = class(b)$. In variants using sim_B we instead store the normalised vectors $\frac{\overrightarrow{ab}}{\|\overrightarrow{ab}\|}$ In this way, the average-case time complexity of the analogical classifiers is reduced from $O(n^3)$ to $O(n^2 \cdot \log(n))$. A different approach to avoid a cubic time complexity is proposed in [29]. This latter approach, called FADANA, is based on precomputing the analogical dissimilarity for a subset of the training data, and relying on the fact that $diss_A$ satisfies the triangle inequality. Our proposed approach is conceptually simpler, however, as we can rely on off-the-shelf implementations of KD trees, and we do not need to tweak the number of instances for which to precompute the analogical dissimilarity.

5.3. Classification based on salient properties

The classifiers from Section 5.2 use the assumption that directions can be identified in $\mathcal{S}$ that point towards class membership, which is what we need for implementing a fortiori reasoning. In practice, however, many of the dimensions of $\mathcal{S}$ will be irrelevant, i.e. ideally we want to look at directions in a relevant subspace of $\mathcal{S}$. The following classification rule, for instance, looks at direction in a one-dimensional subspace of $\mathcal{S}$:

> **If** x is more scary than most horror films **then** x is a horror film.

We will also consider rules with more than one condition, e.g.:

> **If** x is less sweet and less fruity than most wines **then** x is a savory red wine.

To implement a classifier that uses such rules, we need to identify for each class which are the most appropriate directions and how we should interpret 'most'. As candidate directions, we consider the $2n$ salient directions that were selected in Section 4.2.3. Initially, we will interpret 'most' in a strict way. In particular, we will learn rules with conditions of the form $x_0 <_i y$ and $x_0 >_i y$, where y is the item to be classified and x_0 is another (possibly unlabelled) entity. For example, rather than learning the first rule above, we would learn a rule such as

> **If** x is more scary than *the shining* **then** x is a horror film.

After these rules have been learned, we will soften our interpretation of 'most' to reflect that the more a movie is *scary*, the more likely it is a horror film. We will again focus on binary classification problems only, but the method can be straightforwardly extended to multi-class problems.

To learn the rules, we use a variant of the well-known FOIL algorithm [65]. Crucially, we assume that only the rankings $<_i$ (induced by the $2n$ selected salient directions) are available, i.e. we make no use of the actual conceptual space representation of the films. As in the original version of FOIL, our method generates one rule at a time. Each time a rule is created, the positive examples covered by that rule are deleted from the training data. Following this procedure, new rules are incrementally added until 95% of the positive examples have been covered. Then the algorithm is run a second time, generating rules for the negatives examples, in the same way.

Rules are generated by incrementally adding conditions of the form $x_0 <_i y$ or $x_0 >_i y$. In particular, starting with an empty list of conditions ("if *true* then y belongs to class X"), at each step, we choose the condition that maximizes the weighted information gain:

$$WIG(C) = pos_C \cdot \left(\log_2 \frac{pos_C}{pos_C + neg_C} - \log_2 \frac{pos}{pos + neg} \right)$$

where *pos* and *neg* are the number of positive and negative examples which are covered by the rule that has been constructed so far, while pos_C and pos_C are the number of positive and negative examples which are still covered after

Table 13: Parameters used in the experiments based on the FOIL classifier.

test set	N_1	N_2	N_3
movies	100	500	2500
places - GeoNames	2.5	12.5	62.5
places - Foursquare	2.5	12.5	62.5
places - OpenCYC	6.6	33.3	166.6
wines	0.8	4	20

the condition C is added to that rule ($pos_C \leq pos$ and $neg_C \leq neg$). Rules are considered complete when no improvement in terms of information gain can be made anymore, or when the length of the rule reaches a predefined size; we used a maximum of 5 conditions. The accuracy of each rule is then estimated according to its Laplace accuracy (see [65]), defined as $\frac{pos+1}{pos+neg+2}$, where *pos* and *neg* are again the number of positive and negative examples that are covered by the rule.

The result of this training step is a set of rules that derive conclusions of the form $y \in X$ and a set of rules that derive conclusions of the form $y \notin X$. When rules of both types apply to a given test instance y, FOIL uses a weighted majority process, in which rules are weighted based on their Laplace accuracy. Here, we add a second factor, to encode the principle that a rule with condition $x_0 <_i y$ should receive a greater support if $r_i(y) - r_i(x_0)$ is large, and to avoid discarding the rule completely if the condition is violated but $r_i(x_0)$ is close to $r_i(y)$. In other words, we interpret a condition such as $x_0 <_i y$ as a soft constraint. Specifically, we measure the degree to which the condition $y >_i x_0$ is satisfied as follows:

$$lt(x_0, y, i) = \frac{1}{1 + e^{\frac{r_i(y) - r_i(x_0)}{B}}}$$

where B is a parameter that controls how strict the condition $y >_i x_0$ is to be interpreted. We will refer to FOIL_i to denote the version of our algorithm that uses $B = N_i$. Initially, we considered the values $N_1 = 100$, $N_2 = 500$ and $N_3 = 2500$ for the conceptual space of movies. By choosing a range of values we will be able to analyze how sensitive the results are to the choice of B. Since the value of N_i relates to the number of entities, for the remaining problem domains, we have chosen values that reflect a similar proportion of the total number of considered entities. For classification experiments with places, we will consider GeoNames, Foursquare and OpenCYC separately. Since our conceptual space of places contains 403 entities from GeoNames,

we choose $N_1 = 2.5$ because $\frac{100}{15000} \approx \frac{2.5}{403} \approx 0.006$ and similarly for the other values and other place type taxonomies. For the classification experiments with wines, we will consider 120 wines only as not all of the 330 wines in E_{wine} appear in the taxonomy that we will use. As a result we choose $N_1 = 0.8$ since $\frac{0.8}{120} \approx 0.006$. The values N_i that we will consider are summarized in Table 13. In addition, we will use FOIL_0 to denote the version of our classifier in which $lt(x_0, y, i)$ is replaced by the crisp constraint $y >_i x_0$.

The scores for conditions of the form $x_0 >_i y$ are computed in a similar way. The degree to which a rule is satisfied is defined as the minimum of the degrees to which its conditions are satisfied. When categorising a test instance, each rule receives a score which is the product of its Laplace accuracy and the degree to which it is satisfied for that instance. For rules predicting non-membership, this score is multiplied by -1. To make the final decision, we then assume that the test item belongs to the class iff the sum of the scores of the 5 most accurate rules is positive.

6. Experimental results

Our evaluation consists of two parts. First, in Section 6.1 we will evaluate whether the derived semantic relations are sufficiently accurate to be useful in a classification setting. Then in Section 6.2, we will discuss the outcome of a number of crowdsourcing experiments which aim at evaluating more subjective aspects, such as whether the semantic relations can provide useful explanations. All data needed to replicate the experiments has been made available on a companion website[26], including the PPMI weighted vectors, the MDS representations, the directions interpreting each of the terms, the chosen salient properties and the corresponding clusters of terms.

6.1. Evaluation of classifier performance

The baseline classifiers which we will consider are as follows:

k-NN is a standard k-NN classifier (using majority voting if $k > 1$).

SVM_{MDS} is an SVM classifier with a Gaussian kernel, where the feature vector of each item is given by its coordinate in $\mathcal{S}$. We have used the LIBSVM[27] implementation. Because default values of the parameters

[26] `http://www.cs.cf.ac.uk/semanticspaces/`

[27] `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`

yielded very poor results, we have used a grid search procedure to find the optimal value of the C parameter for every class. To this end, the training data for each class was split into 2/3 training and 1/3 validation. Moreover, to address class imbalance we under-sampled negative training examples, such that the ratio between positive and negative training examples was at least 1/2.

SVM$_{BoW}$ is an SVM classifier where the feature vector of each item contains the PPMI values for every term t from the considered text corpus. Apart from this, we used the same configuration as for SVM$_{MDS}$.

C4.5$_{MDS}$ is a standard C4.5 classifier, where the feature vector of each item is given by its coordinate in $\mathcal{S}$. We have used the implementation from the KEEL project[28], using standard values for all parameters.

C4.5$_{dir}$ is a C4.5 classifier, where the feature vector of an entity e contains the values λ_e^i, corresponding to the orthogonal projections of e onto the salient directions (cf. Section 4.2.3).

6.1.1. Results for place types

To evaluate the classifiers in the domain of place types, we generated a number of classification experiments from the Foursquare, GeoNames and OpenCYC taxonomies. For each of the 9 top-level categories that were used by Foursquare in September 2013[29], we considered the corresponding binary classification problem. In the case of GeoNames, we only used 7 of the 9 categories[30], as for 2 categories too few place types were retained in E_{place}. Finally, from the OpenCYC taxonomy, we derived 93 binary classification problems, corresponding to categories at different levels of the hierarchy[31]. We used 5-fold cross-validation in all experiments.

The results are summarized in Table 14, where we consider conceptual spaces of dimensions 20, 50, 100 and 200. A first important observation is that the results are quite robust w.r.t. the chosen number of dimensions:

[28] `http://www.keel.es`

[29] Arts & Entertainment, College & University, Food, Professional & Other Places, Nightlife Spot, Parks & Outdoors, Shops & Service, Travel & Transport, and Residence

[30] We used: H (stream, lake, ...), L (parks, area, ...), R (road, railroad, ...), S (spot, building, farm, ...), T (mountain, hill, rock, ...), U (undersea), V (forest, heath, ...).

[31] A list of these categories can be found on the companion website.

similar results are obtained for 50, 100 and 200 dimensions, although 20 dimensions is too few for most classifiers. Second, as the classification problems are heavily imbalanced, most methods are able to achieve a similar accuracy score. Differences between the F1 score, on the other hand, are more pronounced. Overall, the best results are obtained by *Col*, Btw_A and $Analog_C$. These methods consistently improve 1-NN, which is the best-performing baseline method. Even though the differences with 1-NN are relatively small, they are statistically significant in the case of OpenCYC. Specifically, for OpenCYC we found

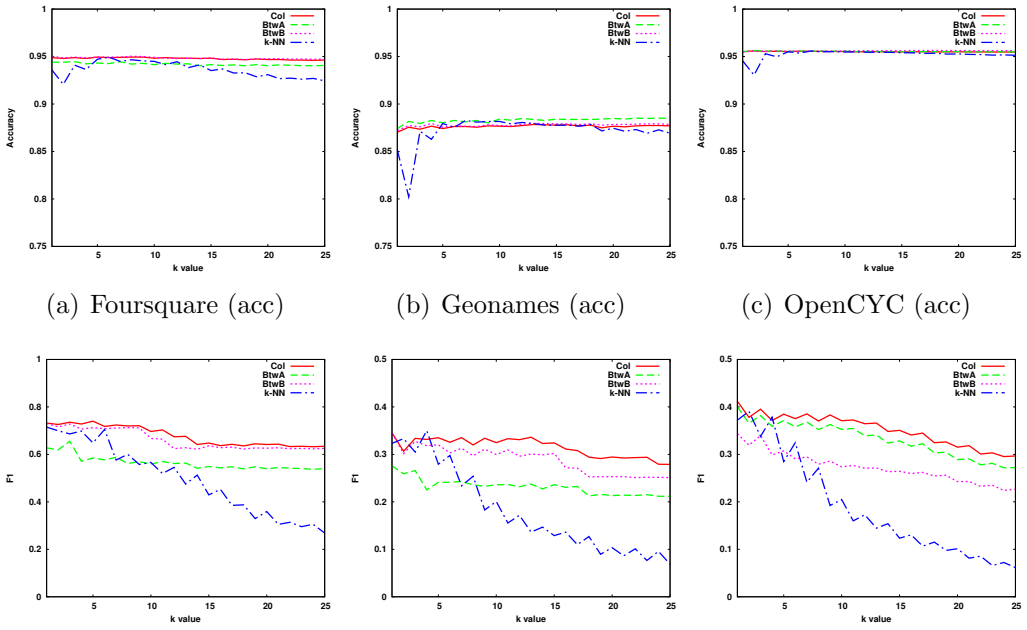
- The accuracy of 1-NN in 50D is significantly improved by *Col* (p-value < 0.0001), Btw_A (p-value < 0.0001) and Btw_B (p-value < 0.0001); the F1 score of 1-NN in 50D is significantly improved by $Analog_C$ (p-value $= 0.0178$).

- The accuracy of 1-NN in 100D is significantly improved by *Col* (p-value < 0.0001), Btw_A (p-value < 0.0001), Btw_B (p-value < 0.0001) and $Analog_B$ (p-value < 0.0001); the F1 score of 1-NN in 100D is significantly improved by *Col* (p-value < 0.0001), Btw_A (p-value $= 0.0008$), $Analog_A$ (p-value $= 0.0018$), $Analog_B$ (p-value $= 0.0033$) and $Analog_C$ (p-value < 0.0001).

- The accuracy of 1-NN in 200D is significantly improved by *Col* (p-value < 0.0001), Btw_A (p-value < 0.0001), Btw_B (p-value < 0.0001) and $Analog_C$ (p-value < 0.0001); the F1 score of 1-NN in 200D is significantly improved by *Col* (p-value < 0.0001), Btw_A (p-value $= 0.0007$), $Analog_B$ (p-value $= 0.0036$) and $Analog_C$ (p-value < 0.0001).

All p-values have been obtained using a two-tailed Wilcoxon signed-rank test. For Foursquare and GeoNames, the number of classification problems (9 and 7 respectively) was not sufficient to achieve statistical significance.

Looking more closely at the results in Table 14, we find that Btw_A outperforms Btw_B. Surprisingly, we also find that *Col* usually performs as good as or better than Btw_A, despite only checking for collinearity. For the analogical classifiers, we find that $Analog_C$ performs better than $Analog_A$ and $Analog_B$, suggesting that a fortiori inference is more reliable for continuous representations than looking for analogical proportions. The FOIL, C4.5 and SVM classifiers are not competitive. However, we do find that SVM_{MDS} outperforms SVM_{BoW}, which suggests that using a conceptual space representation



(a) Foursquare (acc) (b) Geonames (acc) (c) OpenCYC (acc)

(d) Foursquare (F1) (e) Geonames (F1) (f) OpenCYC(F1)

Figure 3: Influence of the value of k on k-NN and the betweenness classifiers

is useful even for standard classifiers. For k-NN and the betweenness classifier, we found $k = 1$ to be a good choice overall. Figure 3 shows how the results are affected by the choice of k. Note that the betweenness classifiers are clearly less sensitive to the choice of k. This suggests that while there may often only be a few sufficiently similar entities that can be exploited by a k-NN classifier, there tend to be many more relevant betweenness triples. In the remainder of this paper, we will only consider the case $k = 1$.

To better understand why the betweenness classifier is able to outperform k-NN, Table 15 gives examples of places which were classified correctly by Btw_A but incorrectly by 1-NN. In many of these examples, the misclassification by 1-NN is because the nearest neighbour is not similar in some important aspect. For example, while *music school* is related to *jazz club* (both being music related venues), these place types have rather different functions (being education and entertainment respectively), which leads 1-NN to misclassify *music school* as an Arts & Entertainment venue. Betweenness, on the other hand, is more demanding, e.g. while *music school* is both similar to *jazz club* and to *piano bar*, *music school* is not located between these

Table 14: Performance of the classifiers for predicting the category of place types from the Foursquare, GeoNames and CYC taxonomies (using an n-dimensional conceptual space induced by MDS) in terms of classification accuracy and F1 measure.

		Foursquare		GeoNames		OpenCYC			Foursquare		GeoNames		OpenCYC	
Algorithm	n	Acc.	F1	Acc.	F1	Acc.	F1	n	Acc.	F1	Acc.	F1	Acc.	F1
Col		0.948	0.692	0.875	0.276	0.953	0.291		0.949	**0.732**	0.870	0.346	0.955	**0.412**
Btw_A		0.947	0.673	0.874	0.284	0.953	0.303		**0.950**	0.730	0.872	0.345	0.955	0.400
Btw_B		0.940	0.562	0.875	0.257	0.952	0.209		0.944	0.628	0.874	0.276	0.955	0.344
$Analog_A$		0.913	0.599	0.808	0.305	0.922	0.280		0.934	0.707	0.830	0.329	0.939	0.403
$Analog_B$		0.931	0.665	0.814	0.311	0.927	0.296		0.938	0.724	0.853	0.345	0.946	0.379
$Analog_C$		0.928	0.648	0.830	0.307	0.939	0.324		0.938	0.724	0.852	0.328	0.945	0.404
$FOIL_0$		0.929	0.575	0.872	0.314	0.948	0.251		0.924	0.500	0.877	0.241	0.950	0.265
$FOIL_1$	20	0.928	0.629	0.865	0.388	0.938	0.311	100	0.928	0.583	0.879	0.340	0.945	0.325
$FOIL_2$		0.930	0.646	0.867	0.396	0.941	0.309		0.930	0.592	0.878	0.340	0.947	0.329
$FOIL_3$		0.928	0.580	0.874	0.360	0.938	0.311		0.928	0.546	0.886	0.317	0.951	0.265
1-NN		0.933	0.668	0.841	0.328	0.942	0.337		0.934	0.715	0.852	0.323	0.945	0.372
$C4.5_{MDS}$		0.936	0.512	0.849	0.163	0.946	0.216		0.905	0.399	0.830	0.172	0.939	0.263
$C4.5_{dir}$		0.917	0.533	0.860	0.274	0.941	0.211		0.913	0.485	0.826	0.229	0.938	0.274
SVM_{MDS}		0.920	0.611	0.815	0.331	0.889	0.275		0.936	0.676	0.876	0.362	0.930	0.355
SVM_{BoW}		0.913	0.358	0.874	0.172	0.946	0.205		0.913	0.358	0.874	0.172	0.946	0.205
Col		0.947	0.717	0.881	0.401	**0.956**	0.383		0.948	0.726	0.875	0.359	0.954	0.397
Btw_A		0.949	0.717	0.883	0.395	**0.956**	0.373		0.947	0.722	0.875	0.350	0.954	0.390
Btw_B		0.943	0.617	0.881	0.349	0.954	0.295		0.945	0.675	0.874	0.318	0.954	0.373
$Analog_A$		0.921	0.636	0.822	0.330	0.933	0.375		0.923	0.670	0.827	0.309	0.940	0.371
$Analog_B$		0.940	0.707	0.853	0.347	0.945	0.382		0.933	0.688	0.852	0.348	0.946	0.375
$Analog_C$		0.925	0.686	0.859	**0.411**	0.942	0.391		0.936	0.687	0.853	0.364	0.946	0.406
$FOIL_0$		0.926	0.564	0.876	0.201	0.950	0.267		0.923	0.501	0.868	0.226	0.949	0.244
$FOIL_1$	50	0.925	0.596	0.860	0.272	0.943	0.329	200	0.918	0.545	0.865	0.295	0.945	0.312
$FOIL_2$		0.926	0.627	0.861	0.285	0.946	0.335		0.920	0.554	0.865	0.293	0.947	0.310
$FOIL_3$		0.928	0.594	0.876	0.300	0.949	0.268		0.921	0.471	0.879	0.219	0.952	0.225
1-NN		0.939	0.710	0.853	0.357	0.945	0.380		0.930	0.677	0.846	0.324	0.945	0.363
$C4.5_{MDS}$		0.925	0.534	0.849	0.178	0.941	0.245		0.914	0.453	0.837	0.198	0.933	0.229
$C4.5_{dir}$		0.918	0.382	0.849	0.374	0.939	0.262		0.912	0.367	0.837	0.277	0.933	0.218
SVM_{MDS}		0.932	0.656	0.859	0.343	0.912	0.328		0.939	0.640	**0.887**	0.309	0.944	0.375
SVM_{BoW}		0.913	0.358	0.874	0.172	0.946	0.205		0.913	0.358	0.874	0.172	0.946	0.205

places. Other examples of misclassifications of this kind include *bike shop* (vs. *bike rental*) and *medical center* (vs. *medical school*). Misclassifications by 1-NN also happen because none of the place types in the training data is sufficiently similar. For example, the place type closest to *veterinarian* is *photography lab* which results in the misclassification of *veterinarian* as a Shops & Services venue. In contrast, betweenness does not require any of the place types to be similar. Because *veterinarian* was identified as being between *animal shelter* and *emergency room*, the betweenness classifier has correctly classified it as belonging to Professional & Other places.

6.1.2. Results for movies

We have evaluated the classifiers in the movies domain on three different types of classes: genres, rating certificates, and plot keywords. Movie genres have been taken from IMDB[32]. We have only considered those 23 genres which have been assigned to at least 100 movies from our data set. Given that multiple genres may be assigned to the same movie, we have considered 23 binary classification problems instead of a single multi-class problem. Second, we considered the task of predicting the rating certificate of movies, focusing on the BBFC[33] certificates and their US equivalent. The ground truth was again obtained from IMDB[34]. The UK ratings can be ranked as follows: *U* < *PG* < *12/12A* < *15* < *18/R18*. To interpret rating prediction as a classification problem, we considered the classes "*PG* or more restrictive", "*12/12A* or more restrictive", "15 or more restrictive" and "*18/R18*". The US ratings can be ranked as *G* < *PG* < *PG-13* < *R/NC-17*, similarly leading to 3 additional classification problems. Finally, we used IMDB plot keywords[35], which are user-defined free text descriptions of movies. We chose the 100 keywords which were most commonly assigned to movies in E_{movie} to define an additional 100 binary classification problems. Note that these genres, ratings and keywords were not considered in the BoW representation of the movies, to allow for a fair evaluation. In practice, however, it would make sense to add the genre labels and keywords to the BoW representation with a high weight, since they tend to be very descriptive.

The results are summarized in Table 16. We have not considered the

[32] `ftp://ftp.fu-berlin.de/pub/misc/movies/database/genres.list.gz`
[33] British Board of Film Classification
[34] `ftp://ftp.fu-berlin.de/pub/misc/movies/database/ratings.list.gz`
[35] `ftp://ftp.fu-berlin.de/pub/misc/movies/database/ratings.list.gz`

Table 15: Examples of places from the Foursquare taxonomy which are misclassified by 1-NN but classified correctly by the betweenness classifier (using Btw_A).

Place	Explanation *Btw*	Category *Btw*	Explanation 1-NN	Category 1-NN
marina	between harbor and plaza	Parks & Outdoor places	similar to pier	Travel & Transport
barbershop	between drugstore and tattoo parlor	Shops & Services	similar to bowling alley	Arts & Entertainment
music school	between auditorium and elementary school	Professional & Other places	similar to jazz club	Arts & Entertainment
campground	between playground and scenic lookout	Parks & Outdoor places	similar to hostel	Travel & Transport
bike shop	between bookstore and motorcycle shop	Shops & Services	similar to bike rental	Travel & Transport
medical center	between fire station and hospital	Professional & Other places	similar to medical school	College & University
legal services	between dojo and financial services	Shops & Services	similar to tech startup	Professional & Other places
spiritual center	between non-profits and synagogue	Professional & Other places	similar to dojo	Shops & Services
cheese shop	between butcher and candy store	Shops & Services	similar to deli	Food
candy store	between grocery store and toy store	Shops & Services	similar to ice cream shop	Food
art gallery	between comedy club and museum	Arts & Entertainment	similar to sculpture garden	Parks & Outdoor places
skate park	between playground and plaza	Parks & Outdoor places	similar to board shop	Shops & Services
veterinarian	between animal shelter and emergency room	Professional & Other places	similar to photography lab	Shops & Services

Table 16: Performance of the classifiers for predicting the genre, rating certificate and keywords of movies in terms of classification accuracy and F1 measure.

		Genres		Ratings		Keywords			Genres		Ratings		keywords	
Algorithm	n	Acc.	F1	Acc.	F1	Acc.	F1	n	Acc.	F1	Acc.	F1	Acc.	F1
$FOIL_0$		0.915	0.517	0.794	0.791	0.878	0.236		0.922	0.558	0.836	0.836	0.883	0.249
$FOIL_1$		0.911	0.530	0.859	0.861	0.878	0.257		0.918	0.575	0.860	0.863	0.882	0.277
$FOIL_2$		0.922	0.540	0.866	0.862	0.903	0.176		0.925	0.581	0.865	0.863	0.902	0.214
$FOIL_3$		0.926	0.405	0.858	0.824	**0.909**	0.02		0.928	0.570	0.861	0.841	**0.909**	0.041
1-NN	20	0.903	0.485	0.829	0.826	0.860	0.216	100	0.903	0.507	0.831	0.831	0.864	0.226
$C4.5_{MDS}$		0.916	0.473	0.827	0.815	0.904	0.112		0.903	0.480	0.807	0.780	0.875	0.195
$C4.5_{dir}$		0.917	0.461	0.828	0.820	0.904	0.102		0.912	0.515	0.824	0.817	0.885	0.199
SVM_{MDS}		0.920	0.558	0.870	0.874	0.862	0.308		0.924	0.624	0.885	**0.887**	0.865	0.357
SVM_{BoW}		0.920	0.607	0.878	0.879	0.860	0.356		0.920	0.607	0.878	0.879	0.860	0.356
$FOIL_0$		0.922	0.544	0.794	0.792	0.882	0.244		0.924	0.568	0.807	0.808	0.887	0.255
$FOIL_1$		0.918	0.564	0.861	0.863	0.880	0.271		0.921	0.589	0.867	0.867	0.883	0.285
$FOIL_2$		0.926	0.576	0.868	0.865	0.903	0.204		0.928	0.599	0.874	0.872	0.903	0.223
$FOIL_3$		0.928	0.463	0.859	0.833	**0.909**	0.029		**0.930**	0.513	0.869	0.848	**0.909**	0.046
1-NN	50	0.904	0.505	0.833	0.832	0.863	0.223	200	0.902	0.501	0.831	0.830	0.864	0.230
$C4.5_{MDS}$		0.907	0.483	0.813	0.810	0.893	0.165		0.896	0.455	0.799	0.798	0.862	0.201
$C4.5_{dir}$		0.917	0.502	0.819	0.807	0.893	0.164		0.908	0.512	0.821	0.816	0.875	0.210
SVM_{MDS}		0.921	0.604	0.883	0.884	0.865	0.344		0.927	**0.633**	**0.886**	**0.887**	0.874	**0.359**
SVM_{BoW}		0.920	0.607	0.878	0.879	0.860	0.356		0.920	0.607	0.878	0.879	0.860	0.356

Table 17: Performance of the classifiers for predicting the category of wines (using an n-dimensional conceptual space induced by MDS) in terms of classification accuracy and F1 measure.

n	20		50		100		200	
Algorithm	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1
Col	0.883	0.525	0.888	0.553	0.882	0.543	0.860	0.488
Btw_A	0.884	0.527	0.888	0.553	0.882	0.543	0.860	0.488
Btw_B	0.883	0.438	**0.891**	0.527	**0.885**	0.538	0.876	0.481
$Analog_A$	0.833	0.505	0.861	0.519	0.865	0.535	0.837	0.462
$Analog_B$	0.869	0.541	0.877	0.548	0.868	0.535	0.847	0.474
$Analog_C$	**0.885**	**0.564**	0.883	**0.570**	0.874	0.554	0.856	0.491
FOIL_0	0.874	0.370	0.860	0.406	0.864	0.315	0.861	0.325
FOIL_1	0.871	0.477	0.856	0.487	0.861	0.375	0.854	0.387
FOIL_2	0.872	0.483	0.859	0.495	0.863	0.386	0.853	0.381
FOIL_3	0.884	0.424	0.866	0.412	0.865	0.357	0.861	0.340
1-NN	0.880	0.559	0.875	0.550	0.869	0.546	0.845	0.461
C4.5_{MDS}	0.852	0.367	0.861	0.365	0.823	0.324	0.846	0.315
C4.5_{dir}	0.868	0.402	0.855	0.405	0.831	0.374	0.861	0.432
SVM_{MDS}	0.839	0.492	0.862	0.516	0.867	**0.564**	**0.880**	**0.495**
SVM_{BoW}	0.874	0.236	0.874	0.236	0.874	0.236	0.874	0.236

betweenness and analogical classifiers here, as they do not scale to the 15000 movies in E_{movie}, due to their quadratic time complexity. Again we find that the results are not very sensitive to the chosen number of dimensions. In contrast to the results for place types, here the SVM classifiers achieve the best performance, followed by the FOIL based methods. The performance of 1-NN and the C4.5 classifiers is not competitive, despite 1-NN being one of the best methods for place types. Nonetheless, it is interesting to see that C4.5_{dir} generally outperforms C4.5_{MDS} suggesting that the interpretable directions may be useful for rule based learners in general. Since the FOIL based methods rely on the most salient properties only, they are most useful for learning common categories such as genres. In Table 16, we indeed find that the difference between SVM_{MDS} and the FOIL based methods is most pronounced for the keywords, which tend to refer to very specific properties. Closer inspection of the results revealed that the relative performance of FOIL_1, compared to SVM_{MDS}, is best for keywords that refer to common movie themes (e.g. *murder*, *police*, *hero*) and worst for keywords that refer to more specific properties (e.g. *new-york-city*, *beach*, *drunkenness*).

6.1.3. Results for wines

To obtain classification problems in the wine domain, we used the taxonomy from `http://winefolly.com/review/different-types-of-wine/`. In total, 122 of the 330 wines in E_{wine} could be matched to wines from that

taxonomy. To generate classification problems, we considered the following 14 categories: Fruity Red, Savory Red, Dry White, Red, White, Sparkling, Tannic, Round, Spicy/Juicy, Blueberry/BlackBerry, Black Pepper Gravel, Smoke Tobacco Leather, Light Citrus Lemon, Medium Perfume Floral. The remaining categories contained too few wines from E_{wine} to be useful. Note that the selected categories are taken from different levels of the taxonomy (e.g. Tannic is a sub-category of Red).

The results, summarized in Table 17, are again not very sensitive to the chosen number of dimensions, but results for 200 dimensions are clearly worse. In contrast, the results for 20 dimensions are close to optimal, whereas 20 dimensions did not lead to competitive results for the place type and movie domaines. Overall, we find that $Analog_C$ achieves the best results, followed by $Col\ /\ Btw_A$, and then 1-NN. In particular $Analog_C$ achieves a higher F1 score than $Col\ /\ Btw_A$ while achieving a similar accuracy (especially in 20D and 50D). On the other hand, $Col\ /\ Btw_A$ achieves a higher accuracy than 1-NN while achieving a similar F1 score (especially in 100D and 200D). The FOIL based classifiers, C4.5 and SVM are not competitive. Interestingly, however, we again find that $C4.5_{dir}$ outperforms $C4.5_{MDS}$, providing further support for the usefulness of the selected salient dimensions.

6.2. Crowdsourcing evaluation of semantic relations

Where Section 6.1 used classification problems to objectively assess the usefulness of the semantic relations, here we look at some subjective aspects, which we have evaluated using CrowdFlower[36], a crowdsourcing platform which has integrated mechanisms for quality control. Our aim was to better understand to what extent the interpretable directions can be used to describe the difference between two entities (Section 6.2.1) and to what extent conceptual betweenness can be used to provide useful explanations of classification decisions (Section 6.2.2). Finally, in Section 6.2.3 we look at how some of the methods compare against human performance, given that some parts of the taxonomies we used are subjective (e.g. ice cream shops are classified as restaurants on TripAdvisor but as shops in OpenCYC).

6.2.1. Identifying terms to compare two movies

In a first experiment, we compared our descriptions of how movies are semantically related with the supervised method from [13]. The latter method,

[36] `http://www.crowdflower.com`

called the Tag Genome, is based on keywords that users have explicitly assigned to movies, together with a supervised learning process aimed at reducing the sparsity of these assignments and to learn a degree of relevance for terms (rather than just having binary assignments). For movie m and tag t, we will write $TG(m,t)$ for the degree of relevance term t has for movie m according to the Tag Genome. We considered the following four methods to generate descriptions of the form "$movie_1$ is more related to t than $movie_2$":

MDS-all selects the term t maximizing $diff_A(movie_2, movie_1; t)$, as explained in Section 4.2.2 (using the 100-dimensional space).

MDS-salient selects the most similar direction among the 200 salient directions in $\mathcal{S}_{movie}$ that were identified in Section 4.2.3 (using the 100-dimensional space). Once the direction is identified, we choose the term t from the corresponding cluster which has the highest Kappa score among all terms that occur at least once in a review of movie 1.

Tag Genome selects the term t that maximizes $TG(movie_1, t) - TG(movie_2, t)$.

PPMI selects the term t maximizing $diff_B(movie_2, movie_1; t)$.

In the experiment, users were asked which of the four resulting statements they thought best described how $movie_1$ differs from $movie_2$. They were also given the option to respond with "I don't know" or "None of the statements applies". To limit the number of unfamiliar movie pairs, we only considered the top 50 most popular movies (in terms of the number of users who have rated the movie on IMDB), resulting in $50 \cdot 49 = 2450$ movie pairs. Each of these pairs was assessed by at least 5 annotators. The total number of (trusted[37]) annotations was 16170. In 3025 cases, the annotator chose the "I don't know" option. We obtained 2339 annotations in favor of MDS-all, 2393 annotations in favor of MDS-salient, 6563 annotations in favor of Tag Genome, and 789 annotations in favor of PPMI. In the remaining 1060 cases, the annotator indicated that "None of the statements applies".

MDS-all and MDS-salient clearly outperform PPMI, although both methods are outperformed by the Tag Genome. This is unsurprising, given that the Tag Genome consists of terms that have been manually assigned to movies

[37] Annotations from reviewers who fail to correctly answer a sufficient number of test questions are automatically removed.

by users (with weights that have been learned in a supervised way, based on feedback from users). To illustrate the difference between the four methods, Tables 18–21 respectively show pairs of movies for which all annotators preferred MDS-all, for which all annotators preferred MDS-salient, for which all annotators preferred Tag Genome, and for which all annotators preferred PPMI. As can be seen from these tables, the tags provided by the Tag Genome are always relevant, although they do not always reflect the most salient properties. For example, the term '70 mm' to describe Die Hard (Table 21, third row) correctly describes one aspect in which Die Hard differs from Django Unchained (since Die Hard has been released on 70mm film), but few people will consider this the most important property in which the two movies differ. MDS-all and MDS-salient often succeed in finding the most important property, but sometimes fail to identify a good label to describe that property. For example, the salient direction containing 'second viewing' (Table 18, fourth row) also contains terms such as 'intriguing', 'enigmatic' and 'a puzzle', presumably because many reviews suggest that some aspects of the plot may only become clear after a second viewing. This direction thus captures one of the main properties of the movie 'Inception', although the label 'second viewing' is not adequate. Similarly, the salient direction corresponding to 'couple' (Table 18, third row) also contains terms such as 'marriage' and 'affair', which captures one of the main themes of Titanic. As expected, MDS-all typically chooses more specific terms (e.g. dinosaurs), while MDS-salient tries to identify more abstract properties (e.g. tragedy). In the few cases where PPMI was preferred by all annotators, the term that was identified tends to correspond to the name of a character, actor or director.

For the majority of movie pairs (1544/2450), at least one assessor preferred the term from the Tag Genome and at least one other assessor preferred MDS-all or MDS-salient, which suggests that these three methods all tend to identify reasonable properties. However, MDS-all and MDS-salient do not always find an intuitive label to associate with that property: there are 578 movie pairs for which all assessors preferred the Tag Genome term. This problem could be alleviated by incorporating better heuristics (e.g. choosing the term from the selected salient direction that has the highest PPMI value for the target movie) or by using automated cluster labelling based on external sources such as Wikipedia [66], but is unlikely to be avoided entirely due to the unsupervised nature of the process. Hybrid methods may be able to combine the best of both worlds, e.g. based on learning directions in the conceptual space $\mathcal{S}$ for keywords that have been associated with films on

IMDB or from the terms in the Tag Genome.

6.2.2. Assessing the usefulness of explanations

In a second crowdsourcing experiment, we have looked at whether explanations help users to assess the reliability of a classification. Again using CrowdFlower, we presented users with arguments of the following form:

> Knowing: X is somewhat between a paintball field and a ski area
> We conclude: X belongs to the category of Parks & Outdoor places

where categories were taken from the Foursquare taxonomy. To generate such explanations, we have used the betweenness based classifier and k-NN. In the latter case, explanations were of the form (for $k = 2$):

> Knowing: X is similar to a paintball field and a ski area
> We conclude: X belongs to the category of Parks & Outdoor places

Note that in each case, users were not shown which place type was being classified (i.e. we always write 'X'). They were only shown the explanation and the classification decision. Users were given four options: (i) based on the given knowledge, I am confident that the conclusion is correct; (ii) based on the given knowledge, I think the conclusion is more or less plausible; (iii) the given knowledge does not support the conclusion; (iv) I don't know.

In total, for each of the considered classifiers, users were shown 391 statements (i.e. one statement for each of the place types in the Foursquare taxonomy). The statements were obtained by using the same configuration of the classifiers as in Section 6.1, using 5-fold cross validation to select the training data. Each statement, for each classifier, was annotated by at least 5 users. We can then rank the classification decisions for a given classifier according to the percentage of human annotators who indicated that they were confident that the conclusion is correct. More precisely, we rank each classification decision according to the value[38] $\frac{pos}{pos+borderline+neg}$, where *pos*, *borderline* and *neg* are the number of annotators who chose the first, second, and third option respectively. The results are summarised in Figure 4, showing the precision-recall trade-off for different cut-offs of the value $\frac{pos}{pos+borderline+neg}$. Since all

[38]Note that options 2 and 3 are treated equally here, i.e. users indicating that the conclusion is 'more or less plausible' are treated in the same way as users saying that the conclusion is not supported. We experimented with several other scoring functions, including functions which considered 'more or less plausible' as equivalent to 'correct', but similar results were found in all cases.

Table 18: Examples of pairs where MDS-all provides the best description according to all annotators.

movie 1	movie 2	MDS-all	MDS-salient	Tag Genome	PPMI
Jurassic Park	Fight Club	dinosaurs	special effects	spielberg	nedry
Alien	Die Hard	aliens	horror movies	space	nostromo
Titanic	Saving Private Ryan	ship	couple	chick flick	hockley
Inception	Braveheart	human mind	second viewing	heist	nolans
Kill Bill: Vol. 1	Goodfellas	martial arts	slow motion	wuxia	vernita

Table 19: Examples of pairs where MDS-salient provides the best description according to all annotators.

movie 1	movie 2	MDS-all	MDS-salient	Tag Genome	PPMI
Titanic	Die Hard	voyage	tragedy	oscar (best picture)	hockley
Toy Story	Batman Begins	children and adults	animation	kids and family	lightyear
Django Unchained	I Am Legend	outrageous	violence	oscar (best supporting actor)	tarantinos
One Flew Over the Cuckoo's Nest	Gran Torino	patients	mental institution	afi 100	mcmurphy
Batman Begins	Transformers	title role	hero	dark	ducard

Table 20: Examples of pairs where the Tag Genome provides the best description according to all annotators.

movie 1	movie 2	MDS-all	MDS-salient	Tag Genome	PPMI
Transformers	Goodfellas	cgi effects	humans	robots	allspark
Goodfellas	Jurassic Park	gangsters	law	mafia	pileggis
Iron Man	Reservoir Dogs	military	graphics	marvel	obadiah
Back to the Future	Reservoir Dogs	disks	dvd	time travel	bttf
Spider-Man	Schindler's List	psychiatrist	mental institution	super-hero	cleg

Table 21: Examples of pairs where PPMI provides the best description according to all annotators.

movie 1	movie 2	MDS-all	MDS-salient	Tag Genome	PPMI
Inception	Toy Story	meditation	second viewing	confusing	nolans
V for Vendatta	Reservoir Dogs	political	politics	author:alan moore	fawkes
Die Hard	Django Unchained	security	guy	70mm	nakatomi
Iron Man	Batman Begins	communications	soldiers	watch the credits	obadiah
Donnie Darko	Spider-Man	indian movies	songs	time loop	hooda

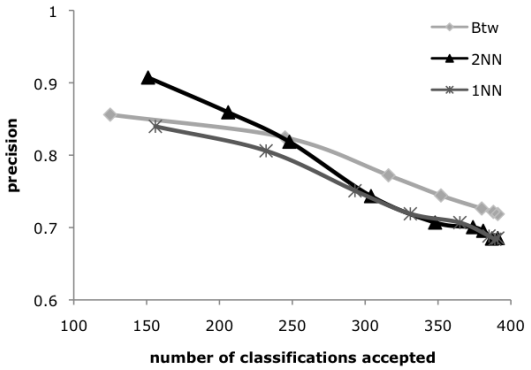


Figure 4: Precision-recall trade-off when only keeping classifications in which at least n human users accept the explanation as being convincing.

three graphs are decreasing, we find that 1-NN, 2-NN and the betweenness classifier (based on the Btw_A measure) indeed generate explanations that help users spot incorrect classifications. Interestingly, 2-NN yields more convincing explanations than 1-NN, despite generating the same classification decisions, i.e. presenting the two most similar place types helps users to better identify misclassifications and borderline cases. Despite having a better classification accuracy overall, the betweenness classifier apparently produces explanations which are slightly less helpful than those provided by 2-NN. This seems related to the fact that the betweenness classifier sometimes makes decisions based on place types which are not similar.

6.2.3. Comparison with human performance

Place type taxonomies are to some extent arbitrary. For example, in the Foursquare taxonomy, 'butcher', 'candy store', 'cheese shop', 'farmers market', 'fish market', 'food court', 'gourmet store' and 'wine shop' are classified under *shops & services* while 'bagel shop', 'bakery', 'cupcake shop','dessert shop', 'donut shop', 'ice cream shop' and 'juice bar' are classified under the disjoint category *food*. Partly this is because the foursquare categories are disjoint (i.e. the taxonomy is a tree). However, in the OpenCYC taxonomy (where categories are not required to be disjoint), we still find arbitrary classifications. For example, 'control tower', 'grain elevator', 'radar station' and 'stable' are classified as buildings, but 'aqueduct', 'radio station' and 'vacant house' are not; 'dockyard', 'bus stop' and 'ski area' are classified as outdoor locations, but 'snowfield' and 'beach' are not. This complicates the inter-

pretation of the experiments in Section 6.1.To better understand how well the classifiers are performing, in a final crowdsourcing experiment we have shown annotators statements of the following form:

> church is a kind of building

Annotators were given the following four options: (i) the statement is correct; (ii) he statement is partially correct; (iii) the statement is incorrect; (iv) I don't know. Assessments where the annotator chose option 4 have been discarded and are not used in the following analysis. Let us write $pos(c,t)$, $borderline(c,t)$ and $neg(c,t)$ for the number of annotators who have respectively chosen option 1, 2 and 3 for an OpenCYC category c and place type t. We have collected judgements for the OpenCYC categories *outdoor location*, *building*, *transport facility*, *tower*, *home*, *business location*, *tourist attraction*, *large building*, *landmark* and *public place*. The place types we have considered are those that belong to the aforementioned categories in OpenCYC, as well as all place types from the Foursquare taxonomy (1230 place types in total). The pairs (c,t) we considered are (i) all pairs where t is a place type from the Foursquare taxonomy, (ii) all pairs where t is a place type from the OpenCYC taxonomy and either t belongs to category c or one of the considered classifiers incorrectly assigned t to category c. This led to a total of 4450 pairs, each of which was assessed by at least 5 annotators.

Table 22 compares the betweenness classifier and 1-NN with human performance on the task of deciding which (category,place type) pairs are correct, where only place types from the OpenCYC taxonomy are considered since there is no ground truth for the place types from Foursquare. The results for Btw_A and 1-NN are based on a 5-fold cross validation. To measure human performance, we have considered three alternatives:

Human strict shows the average performance if for each pair (c,t), we select the response of one of the annotators, and accept the pair as being correct if that annotator has chosen option 1.

Human lenient shows the average performance if we instead accept the pair (c,t) as being correct if that annotator has chosen option 1 or 2.

Human majority shows the performance if we accept all pairs (c,t) for which $pos(c,t) > neg(c,t)$.

The results show that the betweenness classifier is competitive with *human strict* and *human lenient* in terms of accuracy, and even outperforms *human*

Table 22: Comparison between the classifiers and human performance on the task of assigning places from the CYC taxonomy to their categories.

	Accuracy	F1	Precision	Recall
Btw_A	0.671	0.440	0.540	0.371
1-NN	0.591	0.417	0.415	0.419
Human strict	0.675	0.570	0.528	0.620
Human lenient	0.656	0.589	0.504	0.710
Human majority	0.704	0.635	0.557	0.740

strict and *human lenient* in terms of precision. However, human performance is much stronger in terms of recall and (as a result) F1. Interestingly, we observe a wisdom-of-the-crowds effect: the consensus approach used by *Human majority* outperforms the expected performance of a single annotator.

One of the possible applications of the betweenness classifier is to merge different taxonomies. In particular, we have considered the task of assigning place types from Foursquare to the 10 categories from the CYC taxonomy that were used before. Given that the betweenness classifier rivals human performance in terms of precision, we could expect that the places it assigns to these categories would be mostly meaningful. To test this hypothesis, we have used the betweenness classifier and 1-NN to assign places from the Foursquare taxonomy to the 10 considered categories from OpenCYC. For this experiment, the entire OpenCYC taxonomy was used as training data. Given the lack of ground truth, Table 23 compares the results to the human assessments. In particular, the table reports how many place types from Foursquare have been assigned to the CYC categories and what was the precision of these assignments, considering three measures of precision:

all: each of the human assessors considered the assignment correct.

some: at least one of the human assessors considered the assignment correct.

majority: the majority of the human assessors considered the assignment correct, i.e. $pos(c,t) > neg(c,t)$.

We find that Btw_A outperforms 1-NN for all precision measures, although 1-NN assigns slightly more place types, i.e. Btw_A is slightly more cautious in assigning place types to the CYC categories.

Table 23: Precision of the assignment of Foursquare place types to the considered CYC categories according to human assessors.

	classifications	some	all	majority
Btw_A	209	0.919	0.278	0.713
1-NN	277	0.888	0.227	0.632

7. Discussion

The results for place types and, to a lesser extent, wines clearly demonstrate the potential of betweenness and analogical classifiers to avoid some systematic errors that are made by k-NN classifiers, and to come up with reasonable decisions when there are no similar entities that can be exploited. However, when sufficient training data is available, as in the movies domain, SVMs substantially outperform the betweenness and analogical classifiers (as well as k-NN), at least when the C parameter is carefully optimized and class imbalance is addressed (SVMs were uncompetitive when default configurations were used). In such domains, the FOIL based classifiers also perform quite well. The poor performance of FOIL in the place type and movie domains suggests that this method requires a sufficiently high number of training items. The relatively small number of place types and wine varieties makes it harder to learn reliable interpretable directions, and to choose the most salient ones. This is most obvious in the wine domain, where directions in spaces of up to 200 dimensions had to be learned from 330 instances.

Despite being outperformed by SVM classifiers, the FOIL based classifiers have a number of significant advantages. Firstly, we can readily derive intuitive explanations from the decisions made by the FOIL classifier, which can help users assess whether they can trust a classification decision. A second advantage of the FOIL based methods is that we can combine them with other sources of structured information in a natural way (e.g. information about the director and actors associated with a film, extracted from natural language or from linked data). Training the FOIL based classifiers is also computationally more efficient than training the SVM classifiers, given that the latter require a grid search for optimizing the C parameter. Finally, the fact that our FOIL classifiers only rely on symbolic, relational information (i.e. rankings of entities) means that we may be able to make reasonable classification decisions, even if no conceptual space representation for the test item can be obtained. Suppose, for example, that we have some information about an upcoming movie, e.g. that it will be "even scarier than

the Shining". From this information alone, the FOIL based classifiers could predict that the movie will likely belong to the horror genre. In contrast, an SVM classifier would not be able to make any predictions before we have a conceptual space representation (i.e. after the movie has been released and enough reviews have become available). This possibility of using qualitative information derived in other ways (e.g. relation extraction from natural language) could prove particularly important for estimating the properties of rare entities, for which we may have insufficient textual information to induce a reliable conceptual space representation. This also relates to a proposal in [67], where a classifier is learned from natural language instructions and a small number of training examples, and to the idea of zero-shot learning, where no training examples are used at all (see e.g. [14]). For example, Wikipedia defines *legal thriller* as[39] "A suspense film in which the major characters are lawyers and their employees". Given that we know how to interpret properties such as *suspense* and keywords such as *lawyer* as directions in the conceptual space of movies, reasonable classification rules for *legal thriller* could be obtained from its natural language definition and the classification rules we already have for *thriller*.

In future work, we will study how the semantic relations could be used to implement more robust forms of logical inference, using the logic from [25] as a starting point. In this way, we can obtain a purely data-driven way to deal with gaps in a knowledge base, which can be effective even when the number of formulas is relatively small, unlike methods which are based on deriving statistical regularities from the knowledge base itself [3, 4, 68]. On the other hand, such an approach is only suitable when the predicates from the knowledge base can be identified with natural language terms.

8. Conclusions

We have shown how semantic relations between entities can be learned in an entirely unsupervised way, based on a relevant text corpus. The central idea is that we can induce a conceptual space from this text corpus, such that spatial relations in the conceptual space correspond to semantic relations between the entities. Whereas existing approaches have mostly used such learned spatial representations for measuring similarity, we have looked at

[39] `http://en.wikipedia.org/wiki/Thriller_(genre)#Sub-genres_in_film`

betweenness and interpretable directions. We have also showed how these semantic relations can be used to implement classifiers based on well-known patterns of commonsense reasoning, especially interpolation and a fortiori reasoning. Experimental results have demonstrated that these classifiers can outperform standard methods such as SVMs, k-NN, and C4.5. Through a number of crowdsourcing experiments, we have provided further support for the usefulness of the derived semantic relations, for describing the relation between two entities, for generating intuitive explanations of classification decisions, and for merging different taxonomies.

Acknowledgements

This work was supported by EPSRC grant EP/K021788/1.

References

[1] J. Bos, K. Markert, When logical inference helps determining textual entailment (and when it doesn't), in: Proceedings of the Second PASCAL Challenge Workshop on Recognizing Textual Entailment, 2006.

[2] R. West, E. Gabrilovich, K. Murphy, S. Sun, R. Gupta, D. Lin, Knowledge base completion via search-based question answering, in: Proceedings of the 23rd International Conference on World Wide Web, 2014, pp. 515–526.

[3] N. Lao, T. Mitchell, W. W. Cohen, Random walk inference and learning in a large scale knowledge base, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2011, pp. 529–539.

[4] R. Speer, C. Havasi, H. Lieberman, Analogyspace: reducing the dimensionality of common sense knowledge, in: Proceedings of the 23rd AAAI Conference on Artificial intelligence, 2008, pp. 548–553.

[5] A. Collins, R. Michalski, The logic of plausible reasoning: A core theory, Cognitive Science 13 (1) (1989) 1–49.

[6] I. Beltagy, C. Chau, G. Boleda, D. Garrette, K. Erk, R. Mooney, Montague meets Markov: Deep semantics with probabilistic logical form, in: Proceedings of the 2nd Joint Conference on Lexical and Computational Semantics, 2013, pp. 11–21.

[7] A. Freitas, J. C. da Silva, E. Curry, P. Buitelaar, A distributional semantics approach for selective reasoning on commonsense graph knowledge bases, in: Proceedings of the 19th International Conference on Applications of Natural Language to Information Systems, 2014, pp. 21–32.

[8] C. d'Amato, N. Fanizzi, B. Fazzinga, G. Gottlob, T. Lukasiewicz, Combining semantic web search with the power of inductive reasoning, in: Proceedings of the 4th International Conference on Scalable Uncertainty Management, 2010, pp. 137–150.

[9] P. Gärdenfors, Conceptual Spaces: The Geometry of Thought, MIT Press, 2000.

[10] M. Nickel, V. Tresp, H.-P. Kriegel, Factorizing YAGO: Scalable machine learning for linked data, in: Proceedings of the 21st International Conference on World Wide Web, 2012, pp. 271–280.

[11] L. A. Galárraga, C. Teflioudi, K. Hose, F. Suchanek, AMIE: Association rule mining under incomplete evidence in ontological knowledge bases, in: Proceedings of the 22nd International Conference on World Wide Web, 2013, pp. 413–422.

[12] P. Viappiani, B. Faltings, P. Pu, Preference-based search using example-critiquing with suggestions, Journal of Artificial Intelligence Research 27 (2006) 465–503.

[13] J. Vig, S. Sen, J. Riedl, The tag genome: Encoding community knowledge to support novel interaction, ACM Transactions on Interactive Intelligent Systems 2 (3) (2012) 13:1–13:44.

[14] A. Kovashka, D. Parikh, K. Grauman, Whittlesearch: Image search with relative attribute feedback, in: IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 2973–2980.

[15] J. Derrac, S. Schockaert, Enriching taxonomies of place types using Flickr, in: Proceedings of the 8th International Symposium on Foundations of Information and Knowledge Systems, 2014, pp. 174–192.

[16] J. Derrac, S. Schockaert, Characterising semantic relatedness using interpretable directions in conceptual spaces, in: Proceedings of the 21st European Conference on Artificial Intelligence, 2014, pp. 243–248.

[17] E. H. Rosch, Natural categories, Cognitive Psychology 4 (3) (1973) 328–350.

[18] D. L. Medin, M. M. Schaffer, Context theory of classification learning, Psychological Review 85 (1978) 207–238.

[19] R. M. Nosofsky, Exemplar-based approach to relating categorization, identification, and recognition, in: F. G. Ashby (Ed.), Multidimensional models of perception and cognition, Hillsdale, NJ, England: Lawrence Erlbaum Associates, 1992.

[20] T. Cover, P. Hart, Nearest neighbor pattern classification, IEEE Transactions on Information Theory 13 (1) (1967) 21–27.

[21] R. Sun, Robust reasoning: integrating rule-based and similarity-based reasoning, Artificial Intelligence 75 (2) (1995) 241–295.

[22] E. Ruspini, On the semantics of fuzzy logic, International Journal of Approximate Reasoning 5 (1991) 45–88.

[23] D. Dubois, F. Esteva, P. Garcia, L. Godo, H. Prade, Similarity-based consequence relations, in: Proceedings of the Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty, 1995, pp. 171–179.

[24] I. Perfilieva, D. Dubois, H. Prade, F. Esteva, L. Godo, P. Hodáková, Interpolation of fuzzy data: Analytical approach and overview, Fuzzy Sets and Systems 192 (2012) 134–158.

[25] S. Schockaert, H. Prade, Interpolative and extrapolative reasoning in propositional theories using qualitative knowledge about conceptual spaces, Artificial Intelligence 202 (2013) 86–131.

[26] M. Sheremet, D. Tishkovsky, F. Wolter, M. Zakharyaschev, Comparative similarity, tree automata, and diophantine equations, in: Proceedings of the 12th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning, 2005, pp. 651–665.

[27] S. Kok, P. Domingos, Statistical predicate invention, in: Proceedings of the 24th international conference on Machine learning, 2007, pp. 433–440.

[28] M. Abraham, D. Gabbay, U. Schild, Analysis of the talmudic argumentum a fortiori inference rule (kal vachomer) using matrix abduction, Studia Logica 92 (2009) 281–364.

[29] L. Miclet, S. Bayoudh, A. Delhay, Analogical dissimilarity: Definition, algorithms and two experiments in machine learning., Journal of Artificial Intelligence Research 32 (2008) 793–824.

[30] H. Prade, G. Richard, B. Yao, Classification by means of fuzzy analogy-related proportions – a preliminary report, in: Proceedings of the International Conference on Soft Computing and Pattern Recognition, 2010, pp. 297–302.

[31] L. Miclet, H. Prade, Handling analogical proportions in classical logic and fuzzy logics settings, in: C. Sossai, G. Chemello (Eds.), European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty, 2009, pp. 638–650.

[32] M. Bounhas, H. Prade, G. Richard, Analogical classification: handling numerical data, in: Proceedings of the 8th International Conference on Scalable Uncertainty Management, 2014, pp. 66–79.

[33] D. Gentner, Structure-mapping: A theoretical framework for analogy, Cognitive Science 7 (2) (1983) 155–170.

[34] C. Krumhansl, Concerning the applicability of geometric models to similarity data: the interrelationship between similarity and spatial density, Psychological Review 5 (1978) 445–463.

[35] S. Padó, M. Lapata, Dependency-based construction of semantic space models, Computational Linguistics 33 (2) (2007) 161–199.

[36] J. Reisinger, R. J. Mooney, Multi-prototype vector-space models of word meaning, in: Proceedings of the 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, 2010, pp. 109–117.

[37] E. Gabrilovich, S. Markovitch, Computing semantic relatedness using wikipedia-based explicit semantic analysis., in: Proceedings of the International Joint Conference on Artificial Intelligence, Vol. 7, 2007, pp. 1606–1611.

[38] G. Salton, A. Wong, C. S. Yang, A vector space model for automatic indexing, Communications of the ACM 18 (11) (1975) 613–620.

[39] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, R. Harshman, Indexing by latent semantic analysis, Journal of the American Society for Information Science 41 (6) (1990) 391–407.

[40] T. F. Cox, M. A. A. Cox, T. F. Cox, Multidimensional Scaling, Chapman & Hall/CRC, 2001.

[41] K. Erk, Representing words as regions in vector space, in: Proceedings of the Thirteenth Conference on Computational Natural Language Learning, 2009, pp. 57–65.

[42] P. D. Turney, Measuring semantic similarity by latent relational analysis, in: Proceedings of the 19th International Joint Conference on Artificial Intelligence, 2005, pp. 1136–1141.

[43] P. D. Turney, Domain and function: A dual-space model of semantic relations and compositions, Journal of Artificial Intelligence Research 44 (2012) 533–585.

[44] S. Schockaert, H. Prade, Interpolation and extrapolation in conceptual spaces: A case study in the music domain, in: Proceedings of the 5th International Conference on Web Reasoning and Rule Systems, 2011, pp. 217–231.

[45] T. Mikolov, W.-t. Yih, G. Zweig, Linguistic regularities in continuous space word representations, in: Proceedings of NAACL-HLT, 2013, pp. 746–751.

[46] S. J. Hwang, K. Grauman, F. Sha, Analogy-preserving semantic embedding for visual object categorization, in: Proceedings of the 30th International Conference on Machine Learning, 2013, pp. 639–647.

[47] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. Hruschka Jr., T. M. Mitchell, Toward an architecture for never-ending language learning., in: Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, 2010, pp. 1306–1313.

[48] O. Etzioni, M. Banko, S. Soderland, D. S. Weld, Open information extraction from the web, Communications of the ACM 51 (2008) 68–74.

[49] F. M. Suchanek, M. Sozio, G. Weikum, SOFIE: A self-organizing framework for information extraction, in: Proceedings of the 18th International Conference on World Wide Web, 2009, pp. 631–640.

[50] J. J. McAuley, J. Leskovec, Hidden factors and hidden topics: understanding rating dimensions with review text, in: Seventh ACM Conference on Recommender Systems, 2013, pp. 165–172.

[51] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, C. Potts, Learning word vectors for sentiment analysis, in: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, 2011, pp. 142–150.

[52] P. D. Turney, P. Pantel, From frequency to meaning: Vector space models of semantics, Journal of Artificial Intelligence Research 37 (2010) 141–188.

[53] J. B. Tenenbaum, V. d. Silva, J. C. Langford, A global geometric framework for nonlinear dimensionality reduction, Science 290 (5500) (2000) 2319–2323.

[54] R. N. Shepard, Stimulus and response generalization: A stochastic model relating generalization to distance in psychological space, Psychometrika 22 (1957) 325–345.

[55] R. M. Nosofsky, Similarity, frequency, and category representations, Journal of Experimental Psychology 14 (1988) 54–65.

[56] D. D. Lee, H. S. Seung, Learning the parts of objects by non-negative matrix factorization, Nature 401 (6755) (1999) 788–791.

[57] P. O. Hoyer, Non-negative matrix factorization with sparseness constraints, Journal of Machine Learning Research 5 (2004) 1457–1469.

[58] J. Cohen, A coefficient of agreement for nominal scales, Educational and Psychological Measurement 20 (1) (1960) 37 – 46.

[59] C. Freksa, Fuzzy relations and cognitive representations, in: R. Seising, E. Trillas, C. Moraga, S. Termini (Eds.), On Fuzziness, Vol. 298 of Studies in Fuzziness and Soft Computing, Springer Berlin Heidelberg, 2013, pp. 177–183.

[60] G. I. Nalbantov, P. J. Groenen, J. C. Bioch, Nearest convex hull classification, Tech. rep., Erasmus School of Economics (ESE) (2006).

[61] J. Laaksonen, Subspace classifiers in recognition of handwritten digits, Ph.D. thesis, Helsinki University of Technology (1997).

[62] M. Gulmezoglu, V. Dzhafarov, A. Barkana, The common vector approach and its relation to principal component analysis, IEEE Transactions on Speech and Audio Processing 9 (2001) 655–662.

[63] H. Prade, G. Richard, Reasoning with logical proportions, in: Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning, 2010, pp. 545–555.

[64] M. Bounhas, H. Prade, G. Richard, Analogical classification: A rule-based view, in: Proceedings of the 15th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, 2014, pp. 485–495.

[65] J. R. Quinlan, Learning logical definitions from relations, Machine learning 5 (1990) 239–266.

[66] D. Carmel, H. Roitman, N. Zwerdling, Enhancing cluster labeling using Wikipedia, in: Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2009, pp. 139–146.

[67] D. Goldwasser, D. Roth, Learning from natural instructions, Machine Learning 94 (2) (2014) 205–232.

[68] T. Rocktäschel, M. Bosnjak, S. Singh, S. Riedel, Low-dimensional embeddings of logic, in: Proceedings of the ACL 2014 Workshop on Semantic Parsing, 2014, pp. 45–49.