Learning Conceptual Spaces with Disentangled Facets

Rana Alshaikh
School of CS & Informatics
Cardiff University, UK
alshaikhr@cardiff.ac.uk

Zied Bouraoui
CRIL - CNRS & Univ Artois
France
zied.bouraoui@cril.fr

Steven Schockaert
School of CS & Informatics
Cardiff University, UK
schockaerts1@cardiff.ac.uk

Abstract

Conceptual spaces are geometric representations of meaning that were proposed by Gärdenfors (2000). They share many similarities with the vector space embeddings that are commonly used in natural language processing. However, rather than representing entities in a single vector space, conceptual spaces are usually decomposed into several facets, each of which is then modelled as a relatively low-dimensional vector space. Unfortunately, the problem of learning such conceptual spaces has thus far only received limited attention. To address this gap, we analyze how, and to what extent, a given vector space embedding can be decomposed into meaningful facets in an unsupervised fashion. While this problem is highly challenging, we show that useful facets can be discovered by relying on word embeddings to group semantically related features.

1 Introduction

Conceptual spaces (Gärdenfors, 2000) are vector space models that are aimed at representing the entities of a given kind (e.g. movies), together with their associated properties (e.g. scary) and concepts (e.g. thrillers). As such, they are similar in spirit to the vector space models that have been proposed in information retrieval (Deerwester et al., 1990) and natural language processing (Turney and Pantel, 2010; Mikolov et al., 2013), but there are also notable differences. First, in the context of conceptual spaces, an explicit distinction is made between the entities from the domain of discourse, which are represented as vectors, and the corresponding properties and concepts, which are represented as regions (e.g. polytopes) or soft regions (e.g. characterized by a Gaussian). Second, conceptual spaces are organised into a set of facets[1], each of which captures a different aspect of meaning. For instance, in a conceptual space of movies, we may have facets such as genre, language, geographic location, etc.

Each facet is associated with its own vector space, which intuitively captures similarity w.r.t. the corresponding facet. For instance, in a conceptual space of movies, the vector space for the budget facet would only capture whether two movies had a similar budget. Most of these facet spaces tend to be low-dimensional (e.g. modelling budget only needs a single dimension). This clearly differentiates them from traditional semantic spaces, which often have hundreds of dimensions. From an application point-of-view, the separation of vector space models into facets is appealing for several reasons. One key advantage is that it allows us to model similarity in a more flexible, and cognitively more plausible way. A related advantage is that the low-dimensional nature of the facet-specific spaces should make it easier to learn from few examples. Finally, the separation into facets can also make conceptual spaces more interpretable. However, the study of conceptual spaces has mostly focused on modelling cognitive and linguistic phenomena, such as metaphor (Gärdenfors, 1996) and vagueness (Douven et al., 2013), with only few works addressing the challenge of learning such representations from data.

Decomposing conceptual spaces into facets is similar to the problem of disentangled representation learning (DRL), which has recently received considerable interest. However, empirical studies suggest that purely unsupervised DRL methods are unlikely to be successful without a strong inductive bias. In fact, Locatello et al. (2018) found that what mostly matters was how such methods are initialized, rather than what particular optimization objective is used. Moreover, much of

[1] These facets are often referred to as *domains* in the context of conceptual spaces. However, we will use the term facets to avoid confusion with domains of discourse.

Proceedings of the 23rd Conference on Computational Natural Language Learning, pages 131–139
Hong Kong, China, November 3-4, 2019. ©2019 Association for Computational Linguistics

the work in DRL has focused on image processing rather than textual data (which is what we use in this paper). Finally, existing work in DRL is focused on learning factors which are uncorrelated. In our setting, however, the different facets are often highly correlated (e.g. natural disaster movies typically have a high budget).

In this paper, we explore a strategy for decomposing a given vector space embedding into separate facet spaces by first determining which interpretable features are modelled by the vector space and then clustering the word vectors corresponding to these features. Despite being intuitive, given that word embeddings are known to group together functionally similar words, we found this strategy to perform poorly in its basic form. First, simply looking for clusters in word embeddings often leads to thematic clusters, e.g. grouping *horror* together with words such as *scary* and *zombie* rather than other genres such as *western* and *drama*. To address this, we explicitly prevent two words from ending up in the same cluster if the features they are modelling are too similar. Second, in most domains, there are one or two central facets which tend to be highly correlated with most of the other facets (e.g. genre in the movie domain). To ensure that the resulting facet spaces are sufficiently different (rather than capturing minor variations of the most central facets), we found it useful to use an iterative approach, where previously found facets are "removed" from the vector space embedding before proceeding to find further facets. With these two modifications, we find that useful facets can indeed be found, which consistently lead to better classification performance compared to the original vector space embedding.

2 Related Work

Conceptual Spaces. A conceptual space (Gärdenfors, 2000) is a vector representation of the entities from some domain, where the dimensions tend to capture salient features. It is usually assumed that the dimensions of a conceptual space can be grouped into semantic domains, or facets. From a cognitive point of view, this grouping is important because it affects how similarity scores are computed. Intuitively, this is because the dimensions from the same facet tend to interact with each other whereas the dimensions from different facets can be considered in isolation. The problem of learning conceptual spaces from data has only received limited attention to date. One exception is the work of Derrac and Schockaert (2015), which we build on in this paper. In their work, textual descriptions of the considered entities are used to find dimensions that model salient semantic features in a given semantic space. For instance, in a semantic space of movies they found dimensions corresponding to features such as *scary*, *horror* and *zombie*. Note that because these features tend to be correlated, the corresponding dimensions are typically not orthogonal in the input semantic space. For this reason, they refer to these dimensions as *interpretable directions*. More recently, (Ager et al., 2018) proposed a post-processing method to fine-tune these interpretable directions. The main challenge which we address in this paper is to group the features that are found by the method from Derrac and Schockaert (2015) into semantically meaningful facets. A supervised variant of this problem was considered by Banaee et al. (2018). Their approach relies on feature selection methods to find subsets of features that are predictive of particular class labels, based on a set of labelled training examples. In contrast, our focus in this paper is on unsupervised methods, as suitable training data is often not available.

Disentangled Representation Learning (DRL). In the last few years, a large number of generative neural network models have been proposed, with variational autoencoders (VAEs) (Kingma and Welling, 2014) and generative adversarial networks (GANs) (Goodfellow et al., 2014) being the best-known examples. The main underlying idea behind these models is that high-dimensional data (e.g. images) can often be described in terms of a much lower-dimensional latent vector space. Each object can thus be compactly described by its latent code, i.e. the corresponding vector in this latent space. The problem of DRL is to learn such a latent vector representation which is such that (groups of) the dimensions of the latent codes correspond to meaningful interpretable factors. A variety of unsupervised and semi-supervised approaches for learning such disentangled representations have been proposed, such as InfoGAN (Chen et al., 2016), which is based on a modification of the loss function for GANs, and β-VAE (Higgins et al., 2017), which instead uses VAEs as the base model. Conceptually, these approaches modify the loss function of a given generative model by insisting that the dimensions of the latent vector space are in some sense independent. In principle, the latent vector spaces learned by DRL methods can be viewed as conceptual spaces. It is unclear, however, whether purely statistical measures of independence can be sufficient for learning semantically meaningful factors. While interesting results have been obtained for particular applications, after a thorough empirical analysis Locatello et al. (2018) concluded that such results were highly sensitive to the random initialization of the neural network models and the value of hyper-parameters. Their results suggest that, in absence of a suitable supervision signal, high-quality factors can only be learned in the presence of a strong inductive bias. Going beyond unsupervised approaches, (Jain et al., 2018) propose a supervised approach for DRL for text. As supervision signal, they use triplets of the form $(s, d, o)_a$ which encode that relative to aspect a, it holds that s and d are more similar than d and o. Then they use a Convolutional Neural Network (CNN) based model to obtain low-dimensional document embeddings for each considered aspect.

3 Decomposing Conceptual Spaces

Let E be a set of entities of some particular type (e.g. movies) for which a vector space embedding is given. In the following, we will write $\mathbf{e} \in \mathbb{R}^n$ for the embedding of entity e. The first step of our approach consists in applying the method from Derrac and Schockaert (2015), which provides us with a set of words F, each corresponding to a feature that can be modelled as a direction in the vector space. For $f \in F$ we write $\mathbf{d_f}$ for the vector characterizing this direction. Formally, this means that $\mathbf{e_1} \cdot \mathbf{d_f} < \mathbf{e_2} \cdot \mathbf{d_f}$ iff e_2 has the feature f to a higher extent than e_1 (e.g. if f denotes the feature *scary*, then this would mean that movie e_2 is scarier than movie e_1). We briefly recall the method from Derrac and Schockaert (2015) in Section 3.1. Our hypothesis is that we can group these features into meaningful facets and that we can represent these facets as subspaces of the given vector space embedding. Section 3.2 discusses our approach for finding these subspaces.

3.1 Identifying Feature Directions

The method proposed in Derrac and Schockaert (2015) aims to finds a set of features F which can be modelled as directions in the given vector space. The input to their method consists of a text description D_e of each entity e, but they assume no other prior knowledge. In particular, each word w which occurs sufficiently frequently in the document collection $\mathcal{D} = \{D_e \,|\, e \in E\}$ is considered as a candidate feature. To determine whether w should be added as a feature, they train a linear SVM classifier to separate the vector representations of the entities e for which w is mentioned in D_e from the vector representations of the other entities. If this SVM classifier is sufficiently accurate[2], they assume that the word w captures a salient feature. The corresponding feature direction is then characterized by the normal vector $\mathbf{d_f}$ of the hyperplane that was learned by the SVM classifier. We will use the notation pos_w to refer to the set of entities from E which are classified as positive. In our experiments, we used logistic regression classifiers instead of SVMs, which we found to perform similarly but were faster to train.

3.2 Finding Facets

Our aim is to group the features from F into meaningful facets. For instance, in the movies domain, we might expect to see facets corresponding to e.g. genre, language and release date. It does not seem possible (nor desirable) to formally define what constitutes a good facet, a typical problem in unsupervised learning. Intuitively, however, a facet should group features which are of the same kind (e.g. genres) and should in some sense be exhaustive (i.e. all genres, rather than a set of features that refer to one or a few particular genres).

Using subspace clustering. The aim of subspace clustering is to decompose a high-dimensional space into the union of lower-dimensional spaces. This problem has found numerous applications, especially in computer vision. One may wonder whether we can learn useful facets by applying subspace clustering to feature directions $\mathbf{d_f}$. Unfortunately, in our initial experiments, this approach did not prove successful. This is illustrated for the movies domain in Table 1, where we used the state-of-the-art SSC-OMP subspace clustering method (You et al., 2016). For this comparison, we first manually grouped the features from F to obtain a gold standard. The first column of the table shows two of the resulting facets: one corresponding to genres and one corresponding to different

[2] Because of class imbalance, they used Cohen's Kappa instead of classification accuracy.

Movies		
Gold standard	**IncHDB**	**SSC-OMP**
blu, ray, cgi, dolby, surround, computer, technology, theaters, theatre, purchased, ordered, purchase, dvds, amazon, bought, copy, audio, disc, edition, widescreen, transfer, digital, print, vhs, discs	**Initial cluster X_i:** audio, disc, dvds, digital, vhs, dolby, technology, discs, computer, version **Top additional features Y_i:** transfer, edition, blu, cgi, ray, widescreen, amazon, extras, awesome, computer, purchased, purchase, buying, surround, price, trailer, included, favorites, theaters, alot, previews, extra, player	blu, arts, disc, purchased, edition, ordered, crime, british, creepy, disturbing, fighting, charm, rent, honestly, reviewers, oscar, personally, excited, questions, budget, england, education, victims, packed, marriage, tense, detail, fell, hell, deeply, culture, situation, accurate, trailers
thriller, comedic, comedy, documentary, comic, satire, documentaries, drama, melodrama, horror,action, adults, animation, crime, fantasy,family, musical, mystery, romance, war, western	**Initial cluster X_i:** thriller, comedic, comedy, documentary, comic, satire, documentaries, humor, humour, cheesy, adaptation, wit, melodrama, campy, parody **Top additional features Y_i:** hilarious, gags, laughs, jokes, slapstick, funniest, thrillers, funnier, suspense, witty, unfunny, amusing, suspenseful, historical, horror, romance, interviews, psychological	horror, thriller, political, charming, funnier, slapstick, documentaries, hilarious, killed, seat, issue, cheesy, gory, mystery, effects, amazon, widescreen, transfer, realistic, relationship, monster, epic, portrayed, glad, premise, hearing, evil, car, formula, decision, violent, villain, gun, goofy, game, teens, garbage, humor, ruin, product, amount, dad, loving, personality, award, folks

Table 1: Comparison of learned facets with gold standard for the movies domain.

media types (which indirectly captures the time period during which a movie was released). The right-most column shows the closest facets that were found with SSC-OMP. As can be seen, these facets are largely non-sensical. For instance, in the first case, words such as *blu* and *disc* are clustered together with semantically unrelated words such as *fighting*, *england* and *accurate*. In the second example, genres such as *horror* and *thriller* are grouped together with unrelated words such as *cheesy*, *widescreen* and *award*. This negative result seems in accordance to the findings from Locatello et al. (2018) that unsupervised disentangled representation learning seems impossible without a strong inductive bias. We also tried several other subspace clustering methods, for a wide range of different configurations, without obtaining better results. Similarly, we experimented with neural approaches for learning disentangled representations directly from the bag-of-words representations of the entities, but again unsuccessfully.

Using word embeddings. These negative results strongly suggest that some kind of external knowledge is needed to find meaningful facets. To this end, we focus on the use of word embeddings, which seems natural given the fact that words of the same kind (e.g. different names of genres) tend to be used in similar contexts, and can thus be expected to have similar word vectors. In particular, our basic approach for identifying facets consists in clustering the word vectors, from some standard pre-trained word embedding model, corresponding to the features in F. One important drawback of this basic strategy, however, is that it often leads to thematic clusters. For instance, while we would want *horror* to be clustered together with other names of genres, when simply clustering word vectors without any further guidance, *horror* may be clustered together with thematically similar words such as *scary* and *zombie*. To avoid such clusters, we rely on the insight that if a and b are thematically similar words (e.g. *horror* and *zombie*) then the corresponding feature directions $\mathbf{d_a}$ and $\mathbf{d_b}$ will also be similar. However, for paradigmatically similar words, such as *horror* and *comedy*, this should not be the case. In other words, two words should intuitively end up in the same clusters if they have similar word vectors but dissimilar feature directions.

While there are many ways to implement this intuition, we found that using the cosine similarity between $\mathbf{d_a}$ and $\mathbf{d_b}$ was not always reliable. Instead we rely on the following measure of overlap between the sets pos_a and pos_b:

$$o(a,b) = \min\left(\frac{|pos_a \cap pos_b|}{|pos_a|}, \frac{|pos_a \cap pos_b|}{|pos_b|}\right)$$

The dissimilarity between features a and b from F is then defined as follows:

$$d(a,b) = \begin{cases} 1 - \cos(\mathbf{w}_a, \mathbf{w}_b) & \text{if } o(a,b) \leq \lambda \\ 1 & \text{otherwise} \end{cases}$$

where the overlap threshold λ is a hyperparameter and $\mathbf{w}_f$ denotes the word vector for feature f.

The aim of the clustering step is to find a number of disjoint subsets of F, each of which intuitively corresponds to a facet. We will denote these facets by $X_1, ..., X_k$. To avoid finding redundant facets, we identify them in an incremental fashion. In particular, from the clusters obtained by the clustering algorithm, we only select the single most important one, i.e. the one which is most likely to describe a salient facet. For this purpose, we rank clusters according to the following score:

$$\mathit{score}(X_i) = |\bigcup_{f \in X_i} \mathit{pos}_f| \tag{1}$$

This score reflects the intuition that we prefer clusters with features that are general and diverse, i.e. such that most of the entities would have at least one of the features from the cluster. As will be explained below, after the subspace corresponding to this facet has been determined, we iteratively apply the same method on a reduced vector space to find the next most important facet, until the desired number of facets k has been found.

3.3 Modelling Facets as Subspaces

We model each facet X_i as a linear subspace of the given vector space embedding. To find this subspace, we learn new feature directions $\mathbf{c_f}$ for each $f \in X_i$, which still capture these features but lie in a low-dimensional subspace. In particular, we minimize the following objective:

$$\sum_{e \in E} \sum_{f \in X_i} \log \sigma(\mathbf{c_f e} + b_f) \tag{2}$$

where

$$\mathbf{c_f} = \lambda_1^f \mathbf{a_1^i} + \lambda_2^f \mathbf{a_2^i} + ... + \lambda_r^f \mathbf{a_r^i} \tag{3}$$

with r the desired number of dimensions of the subspace. Note that (2) essentially expresses that for each $f \in X_i$, we want to train a logistic regression classifier with coefficient vector $\mathbf{c_f}$. However, as expressed in (3), rather than learning these coefficient vectors independently, they are constrained such that they can be written as a linear combination of the vectors $\mathbf{a_1^i}, ..., \mathbf{a_r^i}$. The resulting feature directions thus span a subspace of (at most) r dimensions. Let $\mathbf{M}_i \in \mathbb{R}^{r \times n}$ be an orthonormal basis for this subspace. Then $\mathbf{e}^i = \mathbf{M}_i\mathbf{e}$ is the r-dimensional facet-specific embedding of entity e.

There may be some features from F which are not contained in X_i but can nonetheless be modelled well in the resulting subspace (i.e. if they are semantically related to the features in X_i). To identify these features, we apply the method from (Derrac and Schockaert, 2015) to the facet-specific embeddings. We write Y_i to denote the features that were thus identified, beyond the ones from X_i.

Next we determine a null space of the basis $\mathbf{M}_i$, i.e. an $(n-r) \times n$ dimensional matrix $\mathbf{R}_i$ satisfying

$$\mathbf{M}_i \mathbf{R}_i^T = \mathbf{0}$$

This matrix $\mathbf{R}_i$ is a basis for the orthogonal complement of the subspace spanned by $\mathbf{M}_i$. Intuitively, it defines what remains of the vector space embedding after we remove (i.e. project away) the subspace modelling the facet X_i.

Dataset	Entities	Attribute	Classes
Movies	13978	Keywords	100
		Genre	23
		Ratings	6
place-types	1383	Foursquare (Fours.)	9
		Geonames (Geo.)	7
		OpenCYC (OpenC.)	20
Organisation	11800	Country	4
		Headquarter Location(HL)	2
Building	3721	Country	2
		Administrative Location(AL)	2

Table 2: Overview of considered datasets.

To find the remaining facets, we repeat the same procedure, but with two changes. First, the $n - r$ dimensional remainder space is used instead of the original embedding space, i.e. we use $\mathbf{R}_i\mathbf{e}$ as the vector representation of e. Second, the features in $X_i \cup Y_i$ are no longer considered by the clustering algorithm. This process is repeated until the desired number of facets has been found, each time considering an increasingly lower-dimensional remainder space and clustering only those features that are not already modelled in a previously identified facet. Intuitively, by learning the facets in this incremental way, we should be able to avoid finding multiple variants of the same facets.

The middle column of Table 1 shows two of the facets that were found with this approach. Intuitively, these facets are clearly more meaningful than those that were found with SSC-OMP.

4 Experimental Analysis

Methods. We have experimented with two clustering algorithms: agglomerative hierarchical average link clustering and HDBSCAN (Campello et al., 2013). However, in the case of HDBSCAN we noticed that when using overlap-based dissimilarity, we typically ended up without any clusters[3]. For HDBSCAN we therefore used cosine similarity instead. We refer to our method with agglomerative clustering as *IncAgg* and to the variant with HDBSCAN as *IncHDB*. In addition, we considered a variant of the method with agglomerative clustering which relies on cosine similarity instead of the overlap-based dissimilarity (*CosIncAgg*).

[3] Note that HDBSCAN does not cluster all the data points, as it removes data points which are considered as noise.

		Place types			Movies			Organisations		Buildings	
		Fours.	Geo.	OpenC.	KeyW.	Genre	Rating	Country	HL.	Country	AL.
DT-D1	MDS	0.34	0.26	0.26	**0.26**	0.38	0.43	0.67	0.24	0.47	0.47
	IncAgg	**0.45**	**0.30**	**0.30**	0.25	**0.40**	**0.47**	**0.76**	**0.26**	**0.50**	**0.50**
	CosIncAgg	0.45	0.26	**0.30**	0.24	0.38	0.43	0.75	0.23	0.43	0.42
	IncHDB	0.43	0.26	0.28	0.25	0.38	0.40	0.50	0.22	0.46	0.46
	NonIncHDB	0.30	0.20	0.27	0.23	0.34	0.40	0.50	0.20	0.46	0.47
	NonIncAgg	0.33	0.24	0.27	0.23	0.33	0.42	0.40	0.21	0.48	0.47
DT-D3	MDS	0.52	0.27	0.32	**0.27**	**0.43**	0.47	0.70	0.27	0.47	0.46
	IncAgg	**0.58**	**0.34**	**0.34**	**0.27**	0.41	**0.47**	0.77	**0.30**	**0.54**	**0.52**
	CosIncAgg	0.54	0.28	**0.34**	0.25	0.40	0.45	**0.78**	0.26	0.47	0.45
	IncHDB	0.57	0.26	0.31	**0.27**	0.41	0.45	0.70	0.27	0.49	0.50
	NonIncHDB	0.43	0.24	0.27	0.26	0.38	0.44	0.60	0.21	0.48	0.49
	NonIncAgg	0.36	0.30	0.29	0.24	0.38	0.45	0.65	0.22	0.51	0.50
SVM	MDS	0.65	0.31	0.35	0.25	**0.54**	0.54	0.71	**0.26**	0.38	0.39
	IncAgg	**0.73**	0.33	**0.37**	**0.26**	**0.54**	**0.55**	0.76	**0.26**	**0.52**	**0.51**
	CosIncAgg	0.62	0.33	0.34	0.25	0.52	0.53	**0.80**	0.12	0.50	0.50
	IncHDB	0.65	0.30	0.36	0.23	0.50	0.51	0.70	0.20	0.51	**0.51**
	NonIncHDB	0.60	**0.35**	**0.37**	0.24	0.46	0.52	0.68	0.24	**0.52**	**0.51**
	NonIncAgg	0.58	**0.35**	0.35	0.24	0.48	0.51	0.72	0.26	0.50	**0.51**
Gaussian	MDS	0.81	0.45	**0.46**	0.26	0.58	0.48	0.74	0.27	0.53	0.51
	IncAgg	**0.87**	**0.48**	0.45	**0.28**	**0.60**	**0.51**	**0.81**	0.27	0.54	**0.55**
	CosIncAgg	0.81	0.45	**0.46**	**0.28**	**0.60**	**0.51**	**0.81**	**0.28**	0.53	0.53
	IncHDB	0.84	0.43	0.43	0.27	**0.60**	**0.51**	0.80	**0.28**	0.54	0.53
	NonIncHDB	0.75	0.41	0.40	0.23	0.51	0.47	0.75	0.27	**0.59**	0.53
	NonIncAgg	0.71	0.46	0.45	0.22	0.52	0.46	0.77	0.27	0.58	0.53

Table 3: Classification tasks performance (in terms of F1 score) when using the MDS space and four variation of the facet-based representations.

Finally, we also report results for variants of our methods in which we did not obtain the facets incrementally (*NonIncAgg* and *NonIncHDB*). In these cases, we simply extract r clusters from the initial set of features F and determine the corresponding facets directly. In all cases, we use 50-dimensional pre-trained GloVe word vectors (Pennington et al., 2014) for clustering the features.

To generate the initial vector space embedding, we follow the approach proposed in (Derrac and Schockaert, 2015) based on multi-dimensional scaling. In all cases, we used 100-dimensional vector spaces and learned 10 facets, each being modelled as a 10-dimensional subspace. To select the set of features F, we initially consider the 500 highest scoring words according to the Kappa metric. However, if we end up without any clusters (in the case of HDBSCAN), we expand the set of features to the 1000 top words. The overlap threshold λ is selected based on held-out tuning data, considering values from $\{0.3, 0.5, 0.7\}$. To flatten the agglomerative clustering, we tune the number of clusters from $\{50, 100, 200\}$[4].

Evaluation tasks. Intrinsic evaluation of the learned facets is difficult, among others because what we might consider to be a natural facet is highly subjective. Therefore, in our quantitative evaluation, we will focus on the impact of the learned facets in a number of classification tasks. This is also motivated by the view that some types of classifiers need semantically meaningful features to perform well. For example, Ager et al. (2018) used low-depth decision trees to evaluate a method for learning feature directions in vector space embeddings. Specifically, if $F = \{f_1, ..., f_m\}$ is the set of features that were identified, then they represent each entity e using the feature vector $(\mathbf{d_{f_1}} \cdot \mathbf{e}, ..., \mathbf{d_{f_m}} \cdot \mathbf{e})$, with $\mathbf{d_f}$ the direction modelling feature f as before. Given that a depth-1 decision tree can only use one of these features, the performance of such a decision tree essentially tells us to what extent the classes that are considered in the supervised classification task have been discovered as features. In our experiments, we will report the result of depth-1 and depth-3 decision trees. As the baseline method, referred to as *MDS*, we will use the top-2000 features that we obtained with the method from Derrac and Schockaert (2015). To evaluate the facets,

[4] The source code is available online at `https://github.com/rana-alshaikh/Disentangled-Facets`.

Cosine similarity	Overlap-based dissimilarity
ORGANISATIONS	
union, europe, protection, aid, spain, right, cross, war, players, black, defenders, german, european	canadian, australian, australia, africa, nations, african, canada, states, countries, asia, united, british, european, competition, world, europe, asian, britain, country, german
PLACE TYPES	
landscapes, serene, tranquil, closeup, surreal, greenery, scenery, scenic, breathtaking, picturesque	sculptures, decoration, churches, fauna, historical, landscapes, st, archeology, small, sculpture, basilica, monuments, convent, heritage, artistic, monument, sacred, forgotten, cemetery, baroque, festival, promenade, renaissance, hall, flora, pavilion, memorial

Table 4: Examples of clusters when standard cosine similarity is used (left) and with the proposed overlap based dissimilarity score (right).

we instead apply this method to find the top-200 features for each of the facet subspaces.

The performance of the decision trees will allow us to evaluate whether we are able to learn higher-quality feature directions thanks to the decomposition of the vector space into facet subspaces. To evaluate the quality of the facets independently of the quality of the feature directions, we also consider classifiers which use as input the facet-specific vector representations $\mathbf{e}^i$ of the entities. Specifically, we train a support vector machine (SVM) for each of the facets, leading to the predictions $p_1, ...p_k$. These predictions are then aggregated to a final prediction using a logistic regression meta-classifier. As baseline, we simply train a single SVM classifier in the full vector space. As our final classifier, we estimate a Gaussian model from the positive training examples. In particular, we estimate a univariate Gaussian for each dimension and multiply the corresponding probabilities. We chose this method because it is sensitive to how well the dimensions of the space are aligned with semantically meaningful properties, and because such Gaussians are commonly used for representing categories in conceptual spaces (Bouraoui and Schockaert, 2018). For the baseline, we use the dimensions of the full vector space. For the facet-based representations, we use the dimensions of the facet subspaces.

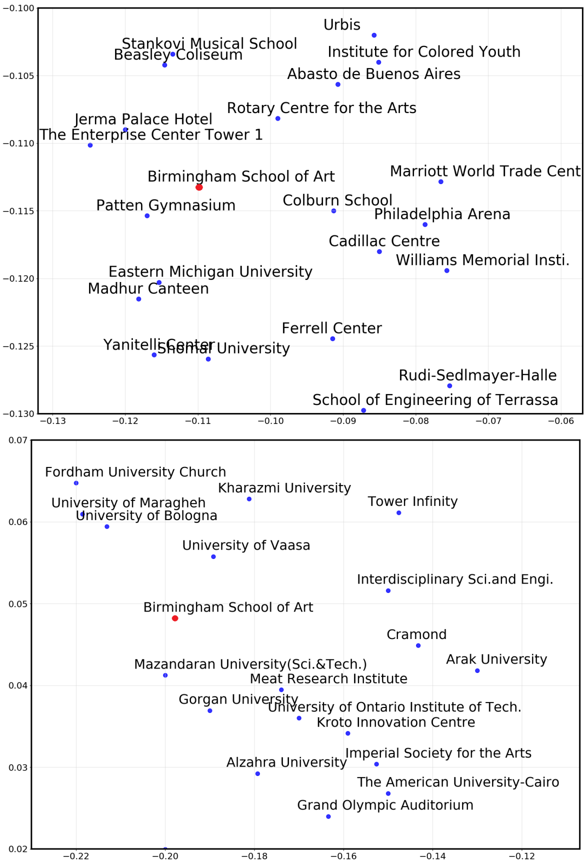

Figure 1: Projection of a 100-dimensional semantic space and 10-dimensional facets of buildings. Top: showing the full space. Bottom: showing the 10-dimensional representations for the facet $X_i = \{$ campuses, students, offices, centers, facilities, area, hotels, homes, bridges, hospitals, cities, shops, stations$\}$

Dataset. We have carried out experiments with vector space embeddings for four different domains. First, we used the *movies* and *place type* domains from (Derrac and Schockaert, 2015), where the embeddings are learned respectively from movie reviews and from Flickr tag co-occurrence distributions. We also considered two additional domains, for which we used Wikipedia articles: *buildings* and *organisations*. In particular, we retrieved all Wikipedia pages whose semantic type on WikiData corresponds to building or organisation. Wikipedia pages containing fewer than 200 words were removed. The bag-of-words (BoW) representation of the remaining Wikipedia concepts were obtained using a standard preprocessing strategy (e.g. removing HTML tags and references), including stopword removal with NLTK (Bird and Loper, 2004). Furthermore, we POS tagged the documents and only retained the nouns and adjectives. Finally, frequent words that occurred in more than 60% of the Wikipedia articles about buildings (resp. organisations) were removed, as well as words that occurred fewer than 10 times. This approach was taken to stay broadly in line with the strategy that was used in (Derrac and Schockaert, 2015). As classification tasks, we used two attributes from WikiData in both domains (being the only attributes for which

BUILDINGS	
Initial cluster X_i:	**Top additional features Y_i:**
architecture, art, history, literature, architectural, society, culture, ancient, scholars, vernacular, classical, historical, contemporary, cultural, medieval	structure, floors, county, architect, graduate, palace, revival, property, hall, united, floor, farm, design, art, space, style, states, downtown, interior, mansion, arena, architectural, architecture, chemistry, entrance.
city, district, town, rural, central, cities, neighborhood, areas, part, section, province, village, north, western, portion, middle, residents, between, branch	company, wife, headquarters, firm, area, events, estate, people, facility, streets, avenue, schools, hotel, commercial, former, states, architects, original, farm, example, residence,

Table 5: Examples of the facets from the Buildings dataset (using *IncAgg* method).

a sufficient number of entities per attribute value was found). The full datasets will be released upon acceptance. The properties of the considered domains and associated classification problems are summarized in Table 2. For each classification problem, we randomly split the labelled examples into 2/3 for training and 1/3 for testing. For tuning we use 5-fold cross-validation over the training set. In the movies domain, where more labelled data is available, we have used fixed splits of 60% for training, 20% for tuning and 20% testing.

Results. The results are summarized in Table 3. Our main method *IncAgg* outperforms the *MDS* baseline for almost all classification tasks and types of classifiers. For the HDBSCAN based variant, the results are more mixed, which seems related to the fact that the overlap based dissimilarity could not be used in that case. Indeed, the cosine based variant of *IncAgg*, i.e. *CosIncAgg*, also performs consistently worse than *IncAgg*. Looking at the performance of *NonIncAgg* and *NonIncHDB* reveals that learning facets in an iterative fashion is critical, given that these two variants perform worse than the baseline in many cases.

Looking more closely at the results of our main method *IncAgg*, it is interesting to note that large improvements are obtained for depth-1 decision trees, which shows that our facet subspaces make it easier to identify features that correspond to the categories from the corresponding classification problems. However, large improvements can also be seen for SVMs, which shows that the actual decomposition of the space is also helpful.

Qualitative Analysis. Figure 1 illustrates how our subspaces capture similarity in a facet-specific way, showing the two first principal components of the embedding of *Birmingham School of Art* in the full space and in the subspace of a facet that intuitively captures building type. While the neighbours in the full space are a mixture of different building types (hotels, commercial buildings, museum, and educational buildings), in the facet subspace all nearest neighbors are universities.

Table 4 illustrates the impact of using overlap-based dissimilarity, where the clusters obtained with cosine similarity are clearly more thematic, while the ones obtained with the overlap-based metric intuitively capture a facet (i.e. geographic location and the natural–cultural opposition). Finally, Table 5 shows some of the facets obtained in the buildings domains. The first example shows a facet which intuitively captures the historical–contemporary opposition, while the second example shows a facet that captures the rural–city opposition.

5 Conclusions

We considered the problem of decomposing a vector space embedding into facets, which are characterized by a set of semantically related features and a corresponding subspace of the embedding. In particular, we focused on unsupervised methods, considering both approaches that rely on the vector space itself (i.e. using subspace clustering) and approaches that additionally take into account the information about word meaning that is captured by pre-trained word vectors. Overall, we found this problem to be highly challenging, in accordance with the findings from Locatello et al. (2018) regarding unsupervised disentangled representation learning. However, we were still able to obtain useful facets based on two crucial modifications to a standard clustering based strategy. First, we measure the similarity between features based on two factors: the similarity between their word vectors and the dissimilarity between their meaning in the vector space embedding (measured in terms of overlap). Second, we found it essential to learn facets in an iterative fashion, to avoid too much redundancy between the different facets.

Acknowledgments

Steven Schockaert was supported by ERC Starting Grant 637277. Zied Bouraoui was supported by CNRS PEPS INS2I MODERN.

References

Thomas Ager, Ondrej Kuzelka, and Steven Schockaert. 2018. Modelling salient features as directions in fine-tuned semantic spaces. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 530–540.

Hadi Banaee, Erik Schaffernicht, and Amy Loutfi. 2018. Data-driven conceptual spaces: Creating semantic representations for linguistic descriptions of numerical data. *Journal of Artificial Intelligence Research*, 63:691–742.

Steven Bird and Edward Loper. 2004. Nltk: the natural language toolkit. In *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*, page 31. Association for Computational Linguistics.

Zied Bouraoui and Steven Schockaert. 2018. Learning conceptual space representations of interrelated concepts. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, pages 1760–1766.

Ricardo JGB Campello, Davoud Moulavi, and Jörg Sander. 2013. Density-based clustering based on hierarchical density estimates. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 160–172.

Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. 2016. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2172–2180.

Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41:391–407.

Joaquín Derrac and Steven Schockaert. 2015. Inducing semantic relations from conceptual spaces: a data-driven approach to plausible reasoning. *Artificial Intelligence*, 228:66–94.

Igor Douven, Lieven Decock, Richard Dietz, and Paul Égré. 2013. Vagueness: A conceptual spaces approach. *Journal of Philosophical Logic*, 42:137–160.

P. Gärdenfors. 2000. *Conceptual Spaces: The Geometry of Thought*. MIT Press.

Peter Gärdenfors. 1996. Mental representation, conceptual spaces and metaphors. *Synthese*, 106:21–47.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. 2017. β-VAE: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*.

Sarthak Jain, Edward Banner, Jan-Willem van de Meent, Iain J Marshall, and Byron C Wallace. 2018. Learning disentangled representations of texts with application to biomedical abstracts. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4683–4693.

Diederik P Kingma and Max Welling. 2014. Auto-encoding variational Bayes. In *Proceedings of the International Conference on Learning Representations*.

Francesco Locatello, Stefan Bauer, Mario Lucic, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. 2018. Challenging common assumptions in the unsupervised learning of disentangled representations. *CoRR*, abs/1811.12359.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543.

Peter D Turney and Patrick Pantel. 2010. From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, 37:141–188.

Chong You, Daniel Robinson, and René Vidal. 2016. Scalable sparse subspace clustering by orthogonal matching pursuit. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3918–3927.