Hierarchical Linear Disentanglement of Data-Driven Conceptual Spaces

Rana Alshaikh[1*], **Zied Bouraoui**[2] and **Steven Schockaert**[1]
[1]Cardiff University, UK
[2]CRIL, Univ Artois & CNRS, France
{alshaikhr,schockaerts1}@cardiff.ac.uk, zied.bouraoui@cril.fr

Abstract

Conceptual spaces are geometric meaning representations in which similar entities are represented by similar vectors. They are widely used in cognitive science, but there has been relatively little work on learning such representations from data. In particular, while standard representation learning methods can be used to induce vector space embeddings from text corpora, these differ from conceptual spaces in two crucial ways. First, the dimensions of a conceptual space correspond to salient semantic features, known as quality dimensions, whereas the dimensions of learned embeddings typically lack any clear interpretation. This has been partially addressed in previous work, which has shown that it is possible to identify directions in learned vector spaces which capture semantic features. Second, conceptual spaces are normally organised into a set of domains, each of which is associated with a separate vector space. In contrast, learned embeddings represent all entities in a single vector space. Our hypothesis in this paper is that such single-space representations are sub-optimal for learning quality dimensions, due to the fact that semantic features are often only relevant to a subset of the entities. We show that this issue can be mitigated by identifying features in a hierarchical fashion. Intuitively, the top-level features split the vector space into domains, allowing us to subsequently identify domain-specific quality dimensions.

1 Introduction

Vector space representations of entities, i.e. entity embeddings, play a central role in information retrieval [Deerwester *et al.*, 1990], natural language processing [Mikolov *et al.*, 2013] and machine learning [Norouzi *et al.*, 2014], among others. Accordingly, a wide variety of approaches have already been proposed for learning such representations, most of which essentially try to learn vectors that represent the similarity structure of the given domain. The use of geometric representations is also common in cognitive science [Shepard, 1957; Nosofsky, 1986; Gärdenfors, 2000], with the conceptual spaces framework by Gärdenfors being a particularly prominent example [Gärdenfors, 2000]. Compared to entity embeddings, conceptual spaces have a much richer structure, allowing them to act as an interface between symbolic and sub-symbolic representations. Essentially, we can think of a conceptual space as being defined by a set of domains, such as e.g. colour, shape or emotion, where each of them is associated with a set of primitive semantic features, called quality dimensions. For instance, the colour domain can be described using the features hue, saturation and intensity. The vector representation of a given domain is then given by the Cartesian product of these quality dimensions. When representing a particular entity in a conceptual space, we need to specify which domains it belongs to, and for each of these domains we need to provide a corresponding vector. We note that conceptual spaces have been used for two different purposes in the literature. On the one hand, they are commonly used in perceptual domains, e.g. for music cognition [Forth *et al.*, 2010; Chella, 2015], where quality dimensions are carefully chosen to maximize how well the resulting conceptual spaces can predict human similarity judgements. On the other hand, they have also been considered as a more general intermediate representation in between neural representations and symbolic ones [Gärdenfors, 1997]. We consider this more general view in this paper, where our aim is add structure to embeddings in the form of semantically meaningful dimensions.

In contrast to conceptual spaces, most methods for learning entity embeddings only aim to capture similarity, and the dimensions of the resulting vector spaces do not typically have any particular meaning. However, despite the similarity-centric nature of most embedding methods, learned vector spaces often exhibit remarkable linear regularities. For example, a well-known property of word embeddings is that many syntactic and semantic relationships can be captured in terms of word vector differences [Mikolov *et al.*, 2013]. We are particularly interested in the fact that important semantic features from a given domain can often be modelled as directions in the corresponding entity embedding. More precisely, for a salient semantic feature f, there often exists a vector $\mathbf{d_f}$ such that $\mathbf{d_f} \cdot \mathbf{a} < \mathbf{d_f} \cdot \mathbf{b}$ tends to hold if b has that feature to a higher extent than a, where we write $\mathbf{a}$ and $\mathbf{b}$ for the vector representations of a and b. Note that only the di-

*Contact Author

rection of $\mathbf{d_f}$ matters in this case, which is why we say that the feature is modelled as a direction rather than as a vector. For example, studying the word vector representations of countries and cities, [Gupta *et al.*, 2015] found directions in the word embedding that highly correlate with (the rankings induced by) properties such as the GDP, life expectancy and military expenditure of a country. In another study, [Kim and de Marneffe, 2013] identified directions that signify adjective scales such as hot–warm–cool–cold in the word embedding. In addition to these supervised approaches, [Derrac and Schockaert, 2015] proposed an unsupervised method which uses text descriptions of the considered entities to identify semantic features that can be characterized as directions. Their core assumption is that words describing semantically meaningful features can be identified by learning for each candidate word w a linear classifier which separates the embeddings of entities that have w in their description from the others. The performance of the classifier for w then tells us to what extent w describes a semantically meaningful feature[1]. This allows learning a linear transformation from the given embedding space to a "disentangled" representation, where entities are represented as vectors whose coordinates correspond to coherent semantic features. The dimensions of this disentangled representation can intuitively be interpreted as the quality dimensions of a conceptual space.

However, many semantic features do not make sense for all entities. For instance, in an embedding of movies, we may consider a feature that captures how closely a movie adheres to the book it is based on. While meaningful for book adaptations, this feature would be non-sensical for other movies. As an important practical implication, if quality dimensions are learned from the full set of entities, while only being sensical for a subset of these entities, we may expect them to be sub-optimal. This problem is illustrated in Fig. 1, which displays a projection of an embedding of organisations. In Fig. 1a, the green dots correspond to those organisations whose associated description contains words such as *political, politic, party, parties, politicians*. Because organisations whose description contains such words are more or less linearly separable from other organisations, the method from [Derrac and Schockaert, 2015] discovered this cluster as a semantic feature. Now consider Fig. 1b, where the yellow dots correspond to organisations whose descriptions contain words such as *democratic* and *left-wings*. While this cluster describes a feature that is intuitively clear (i.e. organisations associated with left-wing political ideas), this feature is only relevant for a subset of organisations (i.e. political ones). A key, and perhaps surprising, observation is that this is reflected in the vector space. In particular, as can be seen in the figure, this feature cannot be characterized well using a single hyperplane.

Decomposing the embedding into different domains could solve this issue, but finding a suitable decomposition is a highly non-trivial problem, especially in unsupervised settings [Locatello *et al.*, 2018; Alshaikh *et al.*, 2019]. Instead of trying to find a hard decomposition of the entity embedding into separate domains, we propose a simple but effective method which is based on applying the method from [Derrac and Schockaert, 2015] in a hierarchical fashion. In the example from Fig. 1, for instance, we can view the feature *political* as defining a domain. To obtain a suitable characterization of the feature *democratic*, it then suffices to apply the method from [Derrac and Schockaert, 2015] to that domain instead of to the full space. In this way, we obtain a set of primary features, and for each of these primary features we obtain a set of sub-features. As confirmed by the experimental results, by learning the features in such a hierarchical way, we obtain semantically more meaningful representations than when directly applying the method from [Derrac and Schockaert, 2015]. This shows that the intuitive hierarchical relationship that exists between features (e.g. the fact that *democratic* is a sub-feature of *political*) is effectively reflected in the structure of the entity embedding. While in general it is well-known that learned embeddings exhibit linear regularities, to the best of our knowledge, this is the first paper to show that these linear regularities have an inherent hierarchical structure.

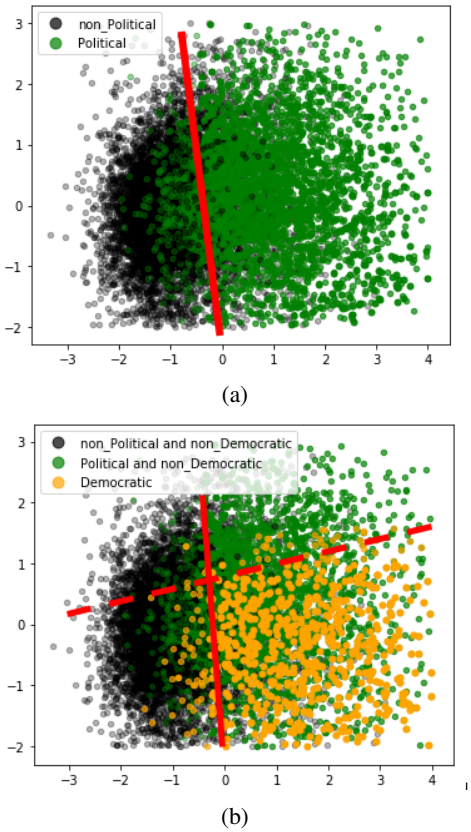

Figure 1: Projection of a 100-dimensional embedding of organisations (see Section 4), showing (a) how organisations that are described with words such as *political, politics,party,parties,politicians* (shown in in green) are separated from others; and (b) how organisations that are described using words such as *democratic, left-wings* (in yellow) are separated from others.

[1]It may seem counter-intuitive to use binary classifiers to learn representations of ordinal features. However, the occurrence or non-occurrence of a word in the description is binary, and this is the most important available signal. We experimented with statistics such as pointwise mutual information, which did not lead to better results.

2 Related Work

An improvement to the method from [Derrac and Schockaert, 2015] was recently proposed in [Ager *et al.*, 2018], where a (non-linear) fine-tuning step was introduced. In [Alshaikh *et al.*, 2019] the problem of clustering the learned dimensions into different facets was studied, finding that this only appears feasible with an external supervision signal, in that case coming from pre-trained word embeddings. Some methods for learning entity embeddings directly incorporate the idea that semantic features should correspond to vector directions. For instance, [Jameel *et al.*, 2017] learn entity embeddings from Wikipedia based on a method that associates an ordinal linear regression model with each word from the vocabulary.

The idea of disentangled representation learning has been widely studied within the context of images. For example, the seminal the InfoGAN model [Chen *et al.*, 2016] uses a variant of generative adversarial networks [Goodfellow *et al.*, 2014] to learn interpretable embeddings of images. Their main idea is to use mutual information to force the individual dimensions of the learned latent representation to correspond to informative properties. When it comes to representations that are learned from text, however, disentangled representation learning has received far less attention. One notable exception is [Jain *et al.*, 2018], where a supervised approach is proposed for learning disentangled document embeddings. As training data, they use triples of the form $(s, d, o)_a$, meaning that relative to aspect a, d is more similar to s than to o. In the context of sentiment analysis, some work has been done on learning representations that disentangle different aspects mentioned in reviews [Ruder *et al.*, 2016]. In [Esmaeili *et al.*, 2019], a general unsupervised disentangled representation learning method is proposed, which is also applied to text. However, this method is only evaluated intrinsically based on statistical measures of disentanglement which do not reveal how semantically meaningful the learned features are. The aforementioned methods focus on representations which are relatively low-dimensional (e.g. typically involving 100 to 300 dimensions). Besides such methods, several approaches have been proposed for learning disentangled text representations which are much higher-dimensional. For example, [Gabrilovich and Markovitch, 2007] learn document vectors with one coordinate for each Wikipedia page, encoding how related the document is to the corresponding Wikipedia entity. Similarly, [Camacho-Collados *et al.*, 2016] proposed entity representations in which dimensions correspond to BabelNet synsets.

3 Identifying Feature Directions

The problem we consider is to derive a *feature-based* (i.e. disentangled) description of a given set of entities. In particular, we want to represent each entity $\mathbf{e}$ as a vector $(f^e_1, ..., f^e_k) \in \mathbb{R}^k$ such that each of its components corresponds to a semantically meaningful feature. This amounts to identifying vectors $\mathbf{d}_1, ..., \mathbf{d}_k$, in the given embedding, which represent the most salient semantic features. The learned vectors will be referred to as *feature directions* to emphasize the fact that only the ordering induced by the dot product $\mathbf{d_i} \cdot \mathbf{e}$ matters. The feature-based representation of $\mathbf{e}$ is simply given by $(\mathbf{e}\cdot\mathbf{d}_1, ..., \mathbf{e}\cdot\mathbf{d}_k)$.

Learning primary features. We first recall the method from [Derrac and Schockaert, 2015] and propose two modifications. This method trains for each word w in the vocabulary a linear classifier which predicts from the embedding of an entity whether w occurs in its description. The words $w_1, ..., w_n$ for which this classifier performs sufficiently well are then used as basic features. To assess classifier performance, Cohen's Kappa score, which can be seen as a correction of classification accuracy to deal with class imbalance, is used. Each of the basic features w is associated with a corresponding vector $\mathbf{d_w}$ (i.e. the normal vector of the separating hyperplane learned by the classifier). These directions are subsequently clustered, which serves to reduce the total number of features. This is useful to keep the representation low-dimensional (e.g. to avoid overfitting when training a classifier on the resulting feature-based representation). By associating features with clusters of words, rather than individual words, features are more likely to correspond to salient properties of the domain. We noticed that irrelevant words tend to be clustered together, resulting in a small number of non-informative clusters, with most of the other clusters corresponding to semantically meaningful features. While [Derrac and Schockaert, 2015] used a variant of k-means, we use affinity propagation [Frey and Dueck, 2007], which gives better results and does not require us to specify the number of clusters, which is crucial in our hierarchical setting.

Once the clusters $C_1, ..., C_k$ are obtained, the final step is to associate with each of them a corresponding feature direction. In [Derrac and Schockaert, 2015], the direction $\mathbf{d_C}$ for a cluster $\{u_1, .., u_m\}$ is defined as the average of $\mathbf{d_{u_1}}, ..., \mathbf{d_{u_m}}$. We instead learn a new linear classifier, which tries to separate entities whose description contains at least one of the words $u_1, ..., u_m$ from the other entities. We then define $\mathbf{d_C}$ as the normal vector of the corresponding hyperplane. This was found to perform better, and it gives us a natural criterion to select entities that have a given feature to a sufficient extent, i.e. those that are classified as positive by this new classifier. For a cluster C, we write pos_C and neg_C for the set of positively and negatively classified entities. We will also refer to the clusters $C_1, ..., C_k$ as the primary feature clusters, and to the associated directions $\mathbf{d_C}$ as the primary feature directions.

Learning sub-features. To find sub-features of a given primary feature f with associated cluster C_f and direction $\mathbf{d_f}$, we essentially re-apply the same method used for learning the primary features, but now only considering the set of entities in pos_{C_f}. In fact, we look for words that linearly separate those entities that are deemed to have the corresponding primary feature[2]. Specifically, for each word w from C_f, we train a linear classifier using only the entities in pos_{C_f}. As before, we use the Kappa score to select those words for which the associated classifier performs sufficiently well. The directions modelling the selected words are then again clustered using affinity propagation. We crucially rely on the fact that this does not require us to specify the number of clusters, as

[2]We also experimented with a variant in which the entities were weighted by the probabilities predicted by a logistic regression classifier. We then trained the sub-feature classifier using a weighted logistic loss, but noticed no consistent improvements.

it would not be feasible to tune the number of sub-clusters for each primary feature. Note that we could recursively repeat the whole process to further refine the sub-features. In this paper, however, we only consider two-level hierarchies, as these were found to be sufficient for the considered datasets.

Let $D_1, ..., D_{k_f}$ be the sub-feature clusters that were found for the primary feature f, and let $\mathbf{d^f_{D_1}}, ..., \mathbf{d^f_{D_{k_f}}}$ be the corresponding sub-feature directions. To define the feature-based representation of a given entity, we now associate one component with each primary feature and one component with each sub-feature. To compute the value corresponding to a given sub-feature for an entity $\mathbf{e}$, we simply take the dot product $\mathbf{e} \cdot \mathbf{d^f_{D_i}}$, i.e. the feature values are computed in exactly the same way as for primary features. Note that while the features are discovered in a hierarchical way, the resulting feature-based representation is thus essentially flat. An alternative would be to combine $\mathbf{e} \cdot \mathbf{d^f_{D_i}}$ with $\mathbf{e} \cdot \mathbf{d_f}$ to define the value of the sub-feature. In initial experiments, however, this was found to perform poorly. One reason is that the classification of entities into pos_{C_f} and neg_{C_f} is not perfect. For instance, an organisation could be a borderline instance of the category of political organisations, but still be a representative example of a left-wing organisation (e.g. a newspaper with a strong left-wing bias), even if in general we tend to think of *left-wing* as a sub-feature of *political*. Another reason is that by simply using the value $\mathbf{e} \cdot \mathbf{d^f_{D_i}}$, we minimize the redundancy between the information captured by the primary feature f and the information captured by this sub-feature.

Inspired by this latter view, we also consider a variant in which we require each sub-feature direction $\mathbf{d^f_{D_i}}$ to be orthogonal to the corresponding primary feature direction $\mathbf{d_f}$, as a way to directly impose the idea that primary features and sub-features should provide complementary information. To this end, we first obtain a sub-feature direction $\mathbf{d^f_{D_i}}$ as before, and then compute the orthogonal decomposition of this vector w.r.t. the vector $\mathbf{d_f}$. In particular, as the final sub-feature direction, we then use the following vector:

$$\tilde{\mathbf{d}}^{\mathbf{f}}_{\mathbf{D_i}} = \mathbf{d^f_{D_i}} - \left(\frac{\mathbf{d^f_{D_i}} \cdot \mathbf{d_f}}{\mathbf{d_f} \cdot \mathbf{d_f}} \right) \mathbf{d_f} \tag{1}$$

4 Evaluation

To evaluate whether the discovered features are semantically meaningful, we test how similar they are to natural categories, by training depth-1 decision trees (meaning that only a single feature can be used for prediction) on our feature-based representations. For instance, in the movie domain, we should expect to see common movie genres among the features. Depth-1 decision trees should thus be able to predict these genres well. Following [Ager *et al.*, 2018], we also evaluate how well natural categories can be characterized using a small set of features, based on the performance of depth-3 decision trees.

Methods. We compare two versions of our method: the standard version (*Sub*) and the version where the orthogonal decomposition (1) is used (*Ortho*). We also consider three baselines. First, we use a model that only uses primary features (*Primary*). Second, we use a model in which feature directions are randomly chosen (*Random*), by sampling each coordinate from a standard normal distribution (which after normalization is equivalent to sampling from a uniform distribution on the hypersphere). Note that while using random directions may seem naive, related methods such as using random projections for dimensionality reduction often perform surprisingly well [Bingham and Mannila, 2001]. As the third baseline, we use average-link Agglomerative Hierarchical Clustering (AHC) to cluster word directions instead of affinity propagation. To obtain a two-level clustering from the dendrogram, we tune distance cut-offs d_1 and d_2 to determine the set of primary clusters and their corresponding sub-clusters. Once the clusters are determined, we learn a corresponding cluster direction in the same way as how the cluster directions for primary features are learned with our method. We also experimented with Hierarchical LDA, but found it too slow to be used on our datasets.

Dataset	Entities	Attributes
Movies	13978	Keywords (100 classes), Genre (23 classes), Ratings (6 classes)
Place-types	1383	Foursquare (9 classes), Geonames (7 classes), OpenCYC (20 classes)
Band	11448	Genres (22 classes), Country of origin (6 classes), Loc. of formation (4 classes)
Organisation	11800	Country (4 classes), Headquarter Loc. (2 classes)
Building	3721	Country (2 classes), Administrative loc. (2 classes)

Table 1: Overview of considered datasets.

Apart from how the feature directions are constructed, the overall number of features also has a strong impact on the result, where increasing the number of features increases the chance that one of them reflects a natural category (even if directions are chosen by chance), but at the risk of overfitting. For the random baseline, we directly tune the number of directions on a held-out development set, considering values from $\{100, 500, 1000, 1500, 2000, 2500\}$. We also verified that no further improvements were possible by choosing more than 2500 or fewer than 100 directions. For the methods which use affinity propagation, we can only influence the number of clusters indirectly, by changing the so-called preference parameter of this clustering algorithm. As is usual, this parameter is chosen relative to the median μ of the affinity scores. For the methods *Sub* and *Ortho*, we considered values from $\{0.7\mu, 0.9\mu, \mu, 1.1\mu, 1.3\mu\}$. For the *Primary* method, we considered a larger set of values to verify that no further improvements were possible: $\{0.5\mu, 0.7\mu, 0.9\mu, \mu, 1.1\mu, 1.3\mu, 1.5\mu\}$. In the case of AHC, to make the results as comparable as possible to those of our model, we tune the cut-offs d_1 and d_2 such that the number of selected clusters is as close as possible to the optimal numbers obtained with affinity propagation. We also tried tuning the cut-off values directly, but this did not give better results.

Datasets. Learning disentangled entity representations only makes sense in domain-specific contexts: while it is natural to think about meaningful features for comparing different movies (e.g. genres), there are few features that would be meaningful in an open-domain setting. To test our method, we focus on five different domains, where for each domain a number of classification problems are considered. The first

		Feat.	Random	AHC	Primary	Sub	Ortho
Place types	Foursquare	D1	0.39±0.01	0.36	0.36	0.43	**0.45**
		D3	0.50±0.01	0.46	0.48	0.54	**0.57**
	Geonames	D1	0.23±0.02	0.22	0.24	0.20	**0.28**
		D3	0.27±0.02	0.29	0.27	0.32	**0.34**
	OpenCYC	D1	0.28±0.01	0.29	0.29	**0.31**	0.30
		D3	0.32±0.01	0.33	0.31	**0.35**	**0.35**
Movies	Keywords	D1	0.24±0.001	**0.26**	**0.26**	0.25	**0.26**
		D3	0.26±0.001	0.27	0.27	**0.28**	**0.28**
	Genres	D1	0.36±0.005	0.38	0.36	**0.43**	0.41
		D3	0.40±0.02	0.43	0.42	0.44	**0.45**
	Ratings	D1	0.44±0.01	0.44	0.45	**0.48**	0.47
		D3	0.46±0.01	0.46	0.47	**0.50**	0.49
Bands	Genres	D1	0.10±0.003	0.16	0.16	**0.17**	0.15
		D3	0.11±0.004	0.14	0.15	**0.16**	0.15
	Country of origin	D1	0.29±0.02	0.33	0.34	**0.40**	0.38
		D3	0.28±0.02	0.33	0.33	**0.43**	0.39
	Loc. of formation	D1	0.13±0.01	0.15	0.14	**0.17**	**0.17**
		D3	0.14±0.01	0.16	0.14	0.16	**0.19**
Org.	Country	D1	0.40±0.01	0.50	0.67	0.66	0.67
		D3	0.44±0.01	0.57	0.65	**0.71**	0.69
	Headquarter loc.	D1	0.17±0.01	**0.23**	0.21	**0.23**	0.22
		D3	0.18±0.01	0.26	0.23	**0.27**	0.25
Buildings	Country	D1	0.53±0.03	0.72	0.74	0.74	0.74
		D3	0.60±0.02	0.75	0.80	**0.81**	0.80
	Adm. loc.	D1	0.27±0.03	0.33	0.37	**0.49**	0.46
		D3	0.29±0.03	0.30	0.31	**0.45**	0.37

Table 2: Performance in terms of F1 score for depth 1 (D1) and 3 (D3) decision trees, considering embeddings learned using MDS.

two domains are obtained from [Derrac and Schockaert, 2015]: a *movies* dataset, where the text descriptions correspond to movie reviews, and a *place types* dataset, where the text descriptions correspond to bags of Flickr tags. In addition, we considered the *organisations* and *buildings* datasets that were introduced in [Alshaikh *et al.*, 2019]. In both cases, the text descriptions correspond to Wikipedia articles and the entities were selected based on Wikidata semantic types. The associated classification tasks similarly correspond to predicting Wikidata attributes of the entities. We used the same process to construct a fifth dataset, focusing on the domain of music bands[3]. Table 1 provides statistics about the datasets.

Entity embeddings. For the movies and place type domains, we used the embeddings that were shared by [Derrac and Schockaert, 2015]. These are 100-dimensional vector spaces obtained using multi-dimensional scaling (MDS), which is a common approach for learning semantic spaces in cognitive science. For the other domains, we generated 100-dimensional embeddings using both MDS and Doc2Vec [Le and Mikolov, 2014], the latter being a popular neural network model for document embedding. In initial experiments, we also considered document embeddings learned by the neural variational document model [Miao *et al.*, 2016]. However, we found the resulting embeddings to be of much lower quality.

Methodology. The datasets are divided into 70% training and 30% testing splits. To tune the parameters, we used 5-fold cross-validation on the training split. Since the movies dataset is substantially larger, in that case we instead used a fixed 60% training, 20% testing and 20% tuning split. We report the results in terms of F1 score. To obtain the feature directions, we used logistic regression and only considered words for which the corresponding Kappa score is at least 0.3. To reduce the computation time, for datasets where this led to more than 5000 features, only the 5000 top-scoring words are retained. When learning directions for the sub-features, we use a lower Kappa score of 0.1, as the corresponding classification problems are often harder (given that sub-features are often about subtle nuances of the primary feature) and the set of candidate words is smaller.

[3]The datasets and source code are available online at https://github.com/rana-alshaikh/Hierarchical_Linear_Disentanglement.

		Feat.	Random	AHC	Primary	Sub	Ortho
Bands	Genres	D1	0.08±0.004	0.09	0.08	**0.09**	0.09
		D3	0.9±0.00	**0.10**	0.09	0.09	0.09
	Country of origin	D1	0.23±0.01	0.22	0.23	**0.24**	**0.24**
		D3	0.23±0.01	0.21	0.24	0.24	**0.26**
	Loc. of formation	D1	0.1±0.003	0.10	0.09	**0.11**	0.11
		D3	0.11±0.002	**0.12**	0.10	0.11	0.10
Org.	Country	D1	0.34±0.01	0.28	0.30	**0.39**	0.37
		D3	0.38±0.02	0.28	0.31	**0.42**	0.42
	Headquarter loc.	D1	0.16±0.02	**0.22**	0.17	**0.22**	**0.22**
		D3	0.18±0.02	0.19	0.19	**0.23**	0.21
Buildings	Country	D1	0.54 ±0.03	0.55	0.56	**0.57**	**0.57**
		D3	0.55±0.01	0.58	0.51	**0.64**	0.60
	Adm. loc.	D1	0.14±0.01	0.15	0.11	**0.17**	0.16
		D3	0.15±0.02	0.13	0.12	**0.16**	**0.16**

Table 3: Performance in terms of F1 score for depth 1 (D1) and 3 (D3) decision trees, with embeddings learned using Doc2Vec.

Experimental results. The results are summarized in Tables 2 and 3. We can see that our *Sub* method clearly outperforms *Primary*. In fact, there are several cases where the performance of *Primary* is comparable with that of *Random*, yet where our *Sub* method performs substantially better (e.g. the depth-1 results for the genres in the movies domain). While the sub-features never perform clearly worse than the primary features, there are some cases where both approaches perform similarly (e.g. the genre attribute in the bands domain). This can be expected when the considered attribute is sufficiently dominant, in which case most or all of the attribute values might correspond to primary features. Interestingly, however, for the movies domain, many of the considered genres were only modelled well as sub-features. The *Ortho* method generally performs well, but slightly worse than *Sub*. Interestingly, however, for the place types, the *Ortho* method performs best overall. This dataset is rather noisy, and taking the orthogonal complement in this case seems to help with preventing overfitting. AHC fails to consistently outperform the other baselines, and is clearly worse than our method. This shows that the improvements obtained by our model are not due to the fact that we impose a hierarchical structure on the features as such, but rather due to specific way in which we learn the directions of the sub-features. Comparing the MDS and Doc2Vec embeddings, we generally see the same trends, although the results for Doc2Vec are consistently worse. To better understand the impact of the number of features, Fig. 2 shows the performance of the different methods in function of the total number of features. As can be seen, while the number of features clearly matters, the improvement that we observed for the sub-features is remarkably robust.

Qualitative analysis. Table 4 shows some illustrative examples of features that were found for the movies domain. As can be seen, sub-features can play a number of different roles. For the primary feature [delightful,cute], the

Primary	Sub-features
[delightful, cute]	[romantic, chemistry, romance], [fantasy, charisma], [disneys],[childish, adventures, fare], [musicals, dance, sings], [magical, magic, enchanting], [heartwarming, sentimental, warm]
[gore, gory]	[bloody, zombie, massacre], [killings, serial, maniac], [hardcore, carnage, gratuitous], [supernatural]
[sci, science]	[creatures, humans, eaten], [franchise, sequel, sequels], [cgi, technology, computer, graphics], [mythology, kingdom, ancient], [doctor, blast, mindless, brain], [scifi, futuristic, outer, aliens], [animation, animators, pixar],
[documentary, interviews]	[athletes], [musicians, concert, concerts, albums], [biography], [individuals, perspective, focuses, individual], [educational], [inspiring, awe, captured, appreciation], [artists, artist], [facts, research, account, thousands], [recording, recordings]

Table 4: Examples of primary and sub features for movies domain. Clusters of words that define a feature are grouped using [...].

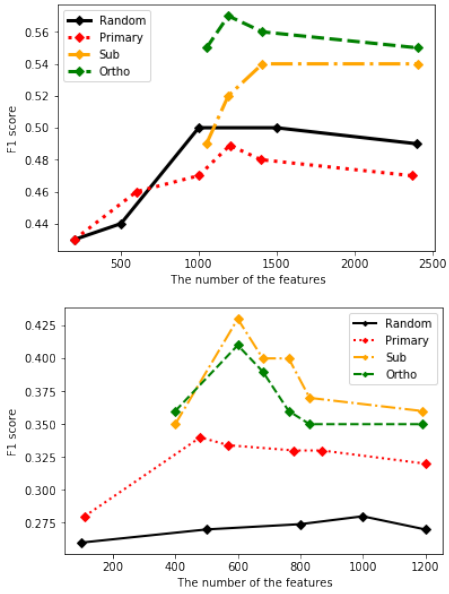

Figure 2: Effect of using more features for the depth-3 decision trees on the Foursquare classification task from the place type domain (top) and the country of origin task from the bands domain (bottom).

sub-features correspond to different kinds of movies that are often described as delightful or cute. These comprise a rather diverse set of movies, encompassing romantic, musicals and children's movies, among others. Without considering sub-features, movies from these various genres would have a very similar representation, which is clearly undesirable. By identifying sub-features, we are able to differentiate between these genres. The primary feature [gore,gory], on the other hand, corresponds to a more coherent movie genre. The sub-features in this case essentially correspond to sub-genres, such as zombie and serial killers movies. For [sci,science] as primary feature, we see the sub-feature [cgi,technology,computer,graphics] as an example of a quality dimension that mostly makes sense within the scope of scifi movies. A similar example is [educational] which is found as a sub-feature of [documentary,interviews]. The examples in Tab. 4 also illustrate the hybrid nature of the found features, where some of them intuitively correspond to more or less well-defined domains while others intuitively correspond to quality dimensions. As far as the primary features are concerned, we would expect to see mostly features of the former kind, but as the example [delightful,cute] illustrates, that is not always the case. One immediate advantage of learning quality dimensions is that we can use them to learn interpretable classifiers. This is illustrated in Fig. 3 that shows depth-3 decision trees for the movie genres *family* and *music*. For most features, we show a subset of the words from the corresponding cluster. It is clear what kind of properties about the considered genres the learned models have uncovered. Such examples show that data-driven conceptual spaces, while clearly being messier than the idealised representations considered by [Gärdenfors, 2000], can indeed bridge between vector space and symbolic representations in a useful way. While these learned representations are not a substitute for the carefully constructed spaces that are often used in perceptual domains, it would be interesting to study in future work to what extent they can be used, for instance, to implement methods for computational creativity based on conceptual spaces [Agres *et al.*, 2015].

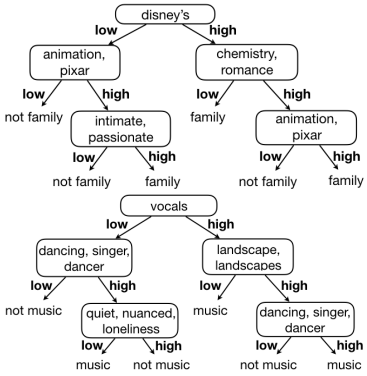

Figure 3: Example of depth-3 decision trees for the *family* and *music* genres in the movies domain, using features from the *Sub* method.

5 Conclusions

Learning feature-based entity representations is complicated by the fact that many features only make sense for particular subsets of entities. We showed that this issue can be mitigated by learning feature representations, characterized as directions in the embedding, in a hierarchical way. Essentially, some of the identified features serve to split up the given entity embedding into different sub-domains, while other features are more similar in spirit to quality dimensions. Compared to strategies that aim to explicitly decompose the embedding into separate domains, our method has the advantage that no hard decisions have to be made about which directions define sub-domains and which directions correspond to quality dimensions. equivalently be seen as semantic features).

Acknowledgments

S. Schockaert was funded by ERC Starting Grant 637277. Z. Bouraoui was funded by ANR CHAIRE IA BE4musIA.

References

[Ager *et al.*, 2018] Thomas Ager, Ondrej Kuzelka, and Steven Schockaert. Modelling salient features as directions in fine-tuned semantic spaces. In *Proc. CoNLL*, pages 530–540, 2018.

[Agres *et al.*, 2015] Kat Agres, Stephen McGregor, Matthew Purver, and Geraint A. Wiggins. Conceptualizing creativity: From distributional semantics to conceptual spaces. In *Proc. ICCC*, pages 118–125, 2015.

[Alshaikh *et al.*, 2019] Rana Alshaikh, Zied Bouraoui, and Steven Schockaert. Learning conceptual spaces with disentangled facets. In *Proc. CoNLL*, 2019.

[Bingham and Mannila, 2001] Ella Bingham and Heikki Mannila. Random projection in dimensionality reduction: applications to image and text data. In *Proc. SIGKDD*, pages 245–250, 2001.

[Camacho-Collados *et al.*, 2016] José Camacho-Collados, Mohammad Taher Pilehvar, and Roberto Navigli. Nasari: Integrating explicit knowledge and corpus statistics for a multilingual representation of concepts and entities. *Artificial Intelligence*, 240:36–64, 2016.

[Chella, 2015] Antonio Chella. A cognitive architecture for music perception exploiting conceptual spaces. In Frank Zenker and Peter Gärdenfors, editors, *Applications of Conceptual Spaces*, pages 187–203. Springer International Publishing, 2015.

[Chen *et al.*, 2016] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Proc. NIPS*, pages 2172–2180, 2016.

[Deerwester *et al.*, 1990] S. Deerwester, S.Dumais, G. Furnas, T. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41:391–407, 1990.

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

[Esmaeili *et al.*, 2019] B. Esmaeili, H. Wu, S. Jain, A. Bozkurt, N. Siddharth, B. Paige, D. H. Brooks, J. G. Dy, and Jan-Willem van de Meent. Structured disentangled representations. In *Proc. AISTATS*, pages 2525–2534, 2019.

[Forth *et al.*, 2010] J. Forth, G. A Wiggins, and A. McLean. Unifying conceptual spaces: Concept formation in musical creative systems. *Minds and Machines*, 20:503–532, 2010.

[Frey and Dueck, 2007] Brendan J Frey and Delbert Dueck. Clustering by passing messages between data points. *Science*, 315:972–976, 2007.

[Gabrilovich and Markovitch, 2007] E. Gabrilovich and S. Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Proc. IJCAI*, pages 1606–1611, 2007.

[Gärdenfors, 1997] Peter Gärdenfors. Symbolic, conceptual and subconceptual representations. In *Human and Machine Perception*, pages 255–270. Springer, 1997.

[Gärdenfors, 2000] Peter Gärdenfors. *Conceptual Spaces: The Geometry of Thought*. MIT press, 2000.

[Goodfellow *et al.*, 2014] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Proc. NIPS*, pages 2672–2680, 2014.

[Gupta *et al.*, 2015] Abhijeet Gupta, Gemma Boleda, Marco Baroni, and Sebastian Padó. Distributional vectors encode referential attributes. In *Proc. EMNLP*, pages 12–21, 2015.

[Jain *et al.*, 2018] Sarthak Jain, Edward Banner, Jan-Willem van de Meent, Iain J Marshall, and Byron C Wallace. Learning disentangled representations of texts with application to biomedical abstracts. In *Proc. EMNLP*, pages 4683–4693, 2018.

[Jameel *et al.*, 2017] S. Jameel, Z. Bouraoui, and S. Schockaert. Member: Max-margin based embeddings for entity retrieval. In *Proc. ACM SIGIR*, pages 783–792, 2017.

[Kim and de Marneffe, 2013] Joo-Kyung Kim and Marie-Catherine de Marneffe. Deriving adjectival scales from continuous space word representations. In *Proc. EMNLP*, pages 1625–1630, 2013.

[Le and Mikolov, 2014] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *Proc. ICML*, pages 1188–1196, 2014.

[Locatello *et al.*, 2018] F. Locatello, S. Bauer, M. Lucic, S. Gelly, B. Schölkopf, and O. Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. *CoRR*, abs/1811.12359, 2018.

[Miao *et al.*, 2016] Yishu Miao, Lei Yu, and Phil Blunsom. Neural variational inference for text processing. In *Proc. ICML*, pages 1727–1736, 2016.

[Mikolov *et al.*, 2013] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Proc. NIPS*, pages 3111–3119, 2013.

[Norouzi *et al.*, 2014] M. Norouzi, T. Mikolov, S. Bengio, Y.Singer, J. Shlens, A. Frome, G.S Corrado, and J. Dean. Zero-shot learning by convex combination of semantic embeddings. In *Proc. ICLR*, 2014.

[Nosofsky, 1986] Robert M Nosofsky. Attention, similarity, and the identification–categorization relationship. *Journal of Experimental Psychology*, 115:39, 1986.

[Ruder *et al.*, 2016] Sebastian Ruder, Parsa Ghaffari, and John G. Breslin. A hierarchical model of reviews for aspect-based sentiment analysis. In *Proc. EMNLP*, pages 999–1005, 2016.

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