*ECAI 2016*
*G.A. Kaminka et al. (Eds.)*

*doi:10.3233/978-1-61499-672-9-1353*

# Entity Embeddings with Conceptual Subspaces as a Basis for Plausible Reasoning

**Shoaib Jameel**[1] and **Steven Schockaert**[2]

**Abstract.** Conceptual spaces are geometric representations of conceptual knowledge in which entities correspond to points, natural properties correspond to convex regions, and the dimensions of the space correspond to salient features. While conceptual spaces enable elegant models of various cognitive phenomena, the lack of automated methods for constructing such representations have so far limited their application in artificial intelligence. To address this issue, we propose a method which learns a vector-space embedding of entities from Wikipedia and constrains this embedding such that entities of the same semantic type are located in some lower-dimensional subspace. We experimentally demonstrate the usefulness of these subspaces as approximate conceptual space representations by showing, among others, that important features can be modelled as directions and that natural properties tend to correspond to convex regions.

## 1 INTRODUCTION

Despite the fact that several large-scale open-domain knowledge bases are now available (e.g. CYC, SUMO, Freebase, Wikidata and YAGO), few knowledge-driven applications rely on logical reasoning. An important reason for this is that available knowledge is often inconsistent. For example, the concept *ice cream shop* is asserted to be disjoint from *restaurant* in CYC, while it is considered a type of *restaurant* on Wikipedia[3]. Another challenge for logical reasoning is that available knowledge is seldom complete. For example, SUMO encodes[4] knowledge about chess, darts and poker, but mentions nothing about checkers.

Humans are remarkably adept at overcoming such challenges [5, 11]. For example, we can recognize that the aforementioned conflict between CYC and Freebase is caused by the vagueness of the categories *restaurant* and *shop*, which both have *ice cream shop* as a borderline case. Similarly, we can deal with knowledge gaps by making inductive inferences, e.g. assuming that properties which hold for chess, darts and poker should hold for checkers as well. Automating such forms of plausible reasoning has proven challenging, among others because they rely on an underlying notion of similarity, which is difficult to characterize using purely symbolic methods.

The solution offered by the theory of conceptual spaces [13] is to represent concepts as regions in a suitable metric space. The points of this space correspond to (actual or possible) entities of a given semantic type, such that similar entities are located close to each other. It is furthermore posited that most natural properties correspond to convex regions, in accordance with prototype theory [26]. Furthermore, the dimensions of a conceptual space correspond to the salient features of the considered domain. For example, a conceptual space of wines could have dimensions relating to sweetness, acidity, fruitiness, amount of tannins, etc. Using conceptual space representations, many cognitive phenomena, including vagueness and induction, can be modelled in a natural way [13, 8, 27, 19]. However, existing applications have focused on a few particular domains in which conceptual space representations can be derived from available metric information. For example, several authors have considered conceptual spaces for music perception [12, 4]. In such cases, the definition of the conceptual space, and its relationship to e.g. audio signals, relies on well-understood insights from the field of music cognition.

The research question we consider in this paper is whether we can automatically obtain approximate conceptual space representations for a wide range of domains, by combining information found in existing knowledge bases with representations derived from large text corpora such as Wikipedia.

Our approach builds on existing work for learning word embeddings from text corpora. Similar to conceptual spaces, word embeddings [21, 24, 29] represent the meaning of words in a high-dimensional Euclidean space, typically as vectors[5]. There are, however, two important differences between word embeddings and conceptual spaces. First, while word embeddings represent all words in a single vector space, conceptual spaces model the entities (and their properties) of a particular semantic type only (e.g. people and cities would be modelled in separate conceptual spaces). Because of this restriction, conceptual spaces can have dimensions that reflect the salient properties of the underlying domain. This allows us to use conceptual spaces for ranking entities (e.g. a conceptual space of cities should have a dimension corresponding the population, allowing us to rank cities from the least to the most populous), modelling context effects[6], and for describing how two entities or concepts are semantically related (e.g. that the rules of chess are more complex than the rules of checkers). In contrast, the dimensions of a word embedding space are essentially meaningless. Second, conceptual spaces clearly differentiate entities, which are modelled as points, from properties, which are modelled as regions. As a result, conceptual spaces can be used to model that a given entity has a given property or belongs to a given category, to model typicality (e.g. an *ice cream shop* could be located in the region modelling *shop* but to-

[1] Cardiff University, UK, email: JameelS1@cardiff.ac.uk
[2] Cardiff University, UK, email: SchockaertS1@cardiff.ac.uk

[3] `https://en.wikipedia.org/wiki/Category:Types_of_restaurants`

[4] `https://github.com/ontologyportal/sumo/blob/master/Sports.kif`

[5] Two notable exceptions are [9] and [30], where words are represented using densities

[6] The context-dependent nature of similarity is modelled in conceptual spaces by allowing dimensions to be rescaled, depending on the importance of the corresponding property in the given context.

wards the border), and to model semantic relations between different properties and categories.

In [6] an approach was proposed for learning conceptual space representations, which consists of (i) representing each entity of a given semantic type as a bag of words (e.g. each movie is represented as its set of user reviews), (ii) converting that bag of words representation to a vector space representation using multi-dimensional scaling (MDS), and (iii) identifying directions corresponding to salient properties of the considered domain in a post-hoc analysis.

An important limitation of the approach from [6] is that it cannot take advantage of relationships between different conceptual spaces. To address this, in this paper we propose a method that learns a single domain-independent vector space, in which each semantic type corresponds to a particular subspace. In other words, we learn conceptual space representations which are themselves embedded in a higher-dimensional vector space. Among others, this allows us to model semantic type hierarchies (e.g. the conceptual space of humans, in our model, is a subspace of the conceptual space of living things). Furthermore, different conceptual spaces can be aligned by taking into account semantic relations between entities of the corresponding types (e.g. the subspaces representing actors, directors and genres can help to obtain a more accurate representation of movies). Another important limitation of the approach from [6] is that MDS requires a distance matrix whose size is quadratic in the number of entities, which severely limits its scalability. In contrast, our model can easily learn representations for millions of entities.

# 2 RELATED WORK

## 2.1 Word embedding

Word embeddings are vector space representations which are used to model the meaning of words. Several existing models construct a vector for each word by applying some form of matrix factorization to a term-term co-occurrence matrix; see [29] for an overview of such approaches. Recently, a number of models have been proposed which instead explicitly optimize the predictive power of the word vectors. For example, the popular Skip-gram model [21] tries to find word vectors that can be used to predict the probability of seeing a context word, given an occurrence of the word being modelled, while the related continuous bag-or-words (CBOW) model focuses on the probability of seeing the word being modelled, given the occurrence of a context word.

An interesting property of word embeddings is that they often capture several kinds of semantic relations, beyond simple similarity. For example, in [21] it is shown that analogical proportions of the form $a$ is to $b$ what $c$ is to $d$ correspond to approximate parallelograms in the space obtained by Skip-gram. They also found that vector addition sometimes corresponds to a form of semantic composition, e.g. adding the vectors for *Germany* and *capital* resulted in a vector which is close to the vector for *Berlin*.

The fact that the vector space obtained by the Skip-gram model satisfies such linear regularities is at first glance somewhat surprising. In [24], the authors analyze what characteristics of a word embedding model can explain this effect, and propose a new model, called GloVe, which is explicitly aimed at capturing linear regularities. Since our model will build on GloVe, we briefly review its formulation. The GloVe model relies on a term-term co-occurrence matrix $X = (x_{ij})$, where $x_{ij}$ is the number of times that word $i$ appears in the context of word $j$. For each term $t_i$ in the vocabulary, two word vectors $w_i$ and $\tilde{w}_i$ and a bias $b_i$ are chosen by minimizing the following objective:

$$J = \sum_{i=1}^{V}\sum_{j=1}^{V} f(x_{ij})(w_i \cdot \tilde{w}_j + b_i + b_j - \log x_{ij})^2 \qquad (1)$$

where $V$ is the number of words in the vocabulary. The function $f$ is used to limit the impact of rare terms, whose co-occurrence counts are considered to be noisy. It is defined as follows:

$$f(x_{ij}) = \begin{cases} \left(\frac{x_{ij}}{x_{max}}\right)^{\alpha} & \text{if } x_{ij} < x_{\max} \\ 1 & \text{otherwise} \end{cases} \qquad (2)$$

where $x_{max}$ is a constant which was fixed as 100. Intuitively, $w_i$ reflects the meaning of term $t_i$ while $\tilde{w}_j$ reflects how the occurrence of that term in the context of another term $t_j$ impacts the meaning of $t_j$.

## 2.2 Knowledge graph embedding

Knowledge bases such as Freebase and Wikidata can essentially be seen as collections of (subject, predicate, object) triples, and can thus be encoded as a graph, where nodes correspond to entities and edges are labelled with relation types. Several authors have looked at the problem of automatically expanding such knowledge graphs [7]. Here, we focus on models that rely on embedding knowledge graphs in a vector space, as we will use similar ideas for aligning different conceptual subspaces. The idea of embedding knowledge graphs in a vector space was proposed in [3]. In particular, they propose the model SE, in which each entity $e_i$ is represented as a vector and each relation $r_k$ is represented using two matrices $R_k^{lhs}$ and $R_k^{rhs}$. The constraint they impose is that the following distance should be small for triples $(e_i, r_k, e_j)$ in the knowledge graph and large for other triples:

$$d(R_k^{lhs} e_i, R_k^{rhs} e_j)$$

where $d$ is either the Euclidean or Manhattan distance. An important drawback of this model is that it requires learning a large number of parameters, which was empirically found to lead to underfitting [2]. In [2] a simpler alternative, called TransE, was proposed, which represents each relation as a vector and considers the following scoring function instead:

$$d(e_i + r_k, e_j)$$

Despite the simplicity of this model, it was shown to substantially outperform SE in practice. However, as noted in [32], TransE is mostly suitable for one-to-one relations. To obtain a more faithful modelling of one-to-many, many-to-one and many-to-many relations, the model TransH is proposed. In this model, both a hyperplane $H_k$ and an $(n-1)$ dimensional vector $r_k$ is associated with each relation type (with $n$ the dimension of the embedding space), and the following scoring function is considered:

$$d(e_i^{H_k} + r_k, e_j^{H_k})$$

where $e_i^{H_k}$ and $e_j^{H_k}$ are the orthogonal projections of $e_i$ and $e_j$ on the hyperplane $H_k$. The TransR model, introduced in [20], follows a similar strategy, but instead associates an $m$-dimensional vector $r_k$ and an $m \times n$ matrix $M_k$ with each relation, and uses the following scoring function:

$$d(e_i M_r + r_k, e_j M_r)$$

The underlying idea is to use the TransE model, after projecting the entities onto a relation-specific space. While in general it is not required that $n = m$, this particular choice was used in all experiments. Finally, [20] also proposes a variant CTransR, in which each entities are clustered, and each relation can have a different representation for each cluster.

In our model, the semantic types of entities play a crucial role. One other approach that explicitly takes semantic type into account is [14]. In particular, they add a regularization term to the objective function of existing embedding models to encode the requirement that entities of the same semantic type should be represented using similar vectors, which they formalize based on two manifold learning algorithms. Unfortunately, the scalability of the resulting method is relatively limited.

In [31] a model is proposed that combines word embedding with knowledge graph embedding. In particular, they jointly learn a representation for words, entities and relations, where the word representations are constrained similarly as in the Skip-gram model and the entities and relations are constrained similarly as in the TransE model. The entity and word representations are aligned either based on Wikipedia anchors or based on the entity names. An improvement of this method was proposed in [35], where the alignment is instead based on the text of the Wikipedia article of the entity. Along similar lines, [34] proposes a model in which the objective functions of Skip-gram and TransE are combined. A third component in their objective function allows the model to take into account an external similarity relation, by imposing the requirement that similar terms should have similar vectors. It is shown that the resulting model improves the word embeddings from Skip-gram.

The model we propose in this paper also combines a word based entity embedding component with a knowledge graph embedding component, although our motivation is different. In particular, [31] and [35] add a word based entity embedding component to a knowledge graph embedding model to improve the predictive performance for entities about which little or nothing is included in the knowledge graph. For example, if the knowledge graph contains the fact that entity $a$ is in relation $R$ with entity $b$, and from the word based component we can derive that entity $a'$ is similar to entity $a$, then we can plausibly derive that entity $a'$ might also be in relation $R$ with entity $b$. Intuitively, we can thus view these approaches as using word based entity embedding to add a kind of smoothing to the knowledge graph embedding model. In contrast, our aim is to model how different entities of the same type are related. The kind of semantic relatedness in which we are interested (e.g. modelling that one building is *taller than* another one) is typically not captured by existing knowledge graphs. Intuitively, we use the word based entity embedding component to learn domain-specific vector space representations, and then use a knowledge graph embedding model to align these spaces. This allows us to improve the representation for semantic types about which little information is available in the considered textual descriptions. In this sense, we can view our model as using the knowledge graph embedding component to add a kind of smoothing to the word based entity embedding. Our motivation is somewhat similar in spirit to [34], but that model focuses on word embeddings rather than entity embeddings.

To the best of our knowledge, our model is the first to use semantic type information to learn domain-specific subspaces.

# 3 DESCRIPTION OF THE MODEL

Our aim is to learn a vector-space embedding of a set of entities $E$, in which entities of the same semantic type lie in some lower-dimensional subspace. Let $S$ be the set of all semantic types. For $s \in S$, we write $E_s$ for the set of all entities of type $s$. We furthermore assume that a set of binary relations $R$ is available, and a set $G \subseteq E \times R \times E$ of triples of the form $(e, k, f)$, encoding that entities $e$ and $f$ are in relation $k$. Finally, we assume that for every entity e, a bag of words $W_e$ describing that entity is available. The model we propose has the following form:

$$J = \alpha J_{\mathit{text}} + (1-\alpha)(J_{\mathit{type}} + J_{\mathit{rel}}) + \beta J_{\mathit{reg}} \tag{3}$$

where $\alpha \in [0, 1]$ and $\beta \in [0, +\infty[$ are parameters controlling the relative importance of the different components of the model. Component $J_{\mathit{text}}$ will be used to constrain the representation of the entities based on their textual description, $J_{\mathit{type}}$ will impose the constraint that entities of the same type belong to a particular subspace, $J_{\mathit{rel}}$ will use the relations in $R$ to improve the alignment between these subspaces, and $J_{\mathit{reg}}$ is a regularization component which will allow the model to automatically select the most appropriate number of dimensions for every subspace. We now discuss each of these components in more detail.

## 3.1 Word based entity embedding

From the bag of words representations $W_e$, we want to find a point $p_e \in \mathbb{R}^n$ for each entity $e$ such that similar entities correspond to nearby points and such that salient features can be interpreted as directions in the space. Specifically, let $f$ be a feature of interest, and let $x_i \in \mathbb{R}$ be the value of feature $f$ for entity $e_i$, i.e. $x_i$ reflects how much $e_i$ has feature $f$. Then there should be a vector $w_f \in \mathbb{R}^n$ such that the orthogonal projection $p'_{e_i}$ of the point $p_{e_i}$ on the line $L_f = \{q \mid q = \lambda \cdot w_f, \lambda \in \mathbb{R}\}$ is given by $p'_{e_i} = c_f x_i w_f + b_f$ for $b_f$ and $c_f$ constants in $\mathbb{R}$. In other words, in a coordinate system where $L_f$ coincides with one of the axes, the corresponding coordinate of $p_e$ should be proportional to $x_i$. This requirement is equivalent to[7]:

$$p_{e_i} \cdot w_f = (c_f x_i + b_f) \cdot \|w_f\| \tag{4}$$

Unfortunately, we do not actually know what are the salient features in most domains. Following [6], we therefore use word co-occurrence as a proxy for feature values. In particular, we assume that each word potentially corresponds to a salient feature, and that the number of times a word co-occurs with a given entity reflects how much that entity has the corresponding feature. This leads to the following constraint

$$p_{e_i} \cdot w_j = g(y_{ji}) + b_j \tag{5}$$

where $y_{ji}$ is the number of times word $t_j$ occurs in $W_{e_i}$, $g$ is a monotonic function that maps co-occurrence statistics to feature values, and $b_j$ is a constant. Typically it will not be possible to satisfy the constraint (5) for all entities and all context words. The assumption underlying this model is that the salient features of an entity affect the co-occurrence statistics of many context words, and that the words for which (5) is (approximately) satisfied, in an optimal solution, will therefore be those that are strongly related to important features of the entity $e_i$.

Note that the requirement in (5) closely resembles the constraints that are optimized by the GloVe model. Moreover, as in the GloVe

[7] We are abusing notation here, using $p_{e_i}$ as a notation for the vector $\overrightarrow{0p_{e_i}}$.

model, we can choose $g(y_{ji}) = \log(y_{ji}) - b_i$ and formalize the objective function as a least squares regression problem, weighted such that frequent terms have a stronger impact on the objective function:

$$J^{E}_{\mathit{text}} = \sum_{e_i \in E} \sum_{t_j \in W_{e_i}} f(y_{ji})(p_{e_i} \cdot w_j + b_i + b_j - \log y_{ji})^2$$

where $f$ is defined as in (2). The resulting model is essentially the same as GloVe, but instead of modelling word-word co-occurrence we now model entity-word co-occurrence. The geometric interpretation, however, is different, as we view entities as points and context words as vectors. We can further constrain the word vectors $w_j$ by adding a second component, capturing word-word co-occurrences, which corresponds to the original GloVe model. In particular, we define $J_{\mathit{text}} = J^{E}_{\mathit{text}} + J_{\mathit{glove}}$, where $J_{\mathit{glove}}$ is the objective function $J$ defined in (1).

### 3.2 Subspace constraints

A key distinguishing feature of our model is that all entities of a given type $s$ are imposed to belong to the same subspace. To formalize this constraint, we associate with each semantic type $s$ a set of $n+1$ points $p^s_0, ..., p^s_n \in \mathbb{R}^n$ and express that for each entity $e_i$ of type $s$, the point $p_{e_i}$ can be written as a convex combination of the points $p^s_0, ..., p^s_n$:

$$J_{\mathit{type}} = \sum_{s \in S} \sum_{e \in E_s} \|p_e - \sum_{j=0}^{n} \lambda^{e,s}_j p^s_j\|^2$$

where we impose that $\lambda^{e,x}_j \geq 0$ and $\sum_{j=0}^{n} \lambda^{e,s}_j = 1$. Note that on its own, this component is trivial, as it suffices to choose any set of points $p^s_0, ..., p^s_n$ in general linear position. However, we will additionally require that the space spanned by the points $p^s_0, ..., p^s_n$ is as low-dimensional as possible. In particular, let $M_s$ be the $n \times n$ matrix whose $i^{th}$ row vector is $p^s_i - p^s_0$. Then clearly the rank of $M_s$ is equal to the dimension of the space spanned by $p^s_0, ..., p^s_n$. We now want to add a regularization term to penalize high-rank matrices $M_s$. Unfortunately, no efficient methods exist for directly minimizing the rank of a matrix $M$. The relaxation suggested in [10] is to minimize the nuclear norm $\|M\|_*$ instead (i.e. the sum of the singular values of $M$). This technique was empirically shown to lead to low-rank matrix solutions in many applications, and is known to be equivalent to rank minimization in certain cases [25]. The regularization term associated with $J_{\mathit{type}}$ is thus given by

$$J^1_{\mathit{reg}} = \sum_{s \in S} \|M_s\|_*$$

To implement nuclear norm regularization, we have used the recently proposed method from [16].

We will also consider a variant in which the points $p^s_0, ..., p^s_n$ are additionally required to be close to each other:

$$J^{\mathit{comb}}_{\mathit{type}} = \sum_{s \in S} \big( \sum_{e \in E_s} \|p_e - \sum_{j=0}^{n} \lambda^{e,s}_j p^s_j\|^2 + \sum_{j=0}^{n} d(p^s_j, c^s_j) \big)$$

where $c^s_j = \frac{1}{n+1} \sum_j p^s_j$ is the center-of-gravity of the points $p^s_0, ..., p^s_n$.

### 3.3 Modelling relations

Often we have information about how entities of different types are related, e.g. the fact that Steven Spielberg is the director of Jurassic Park. Such relationships can help us to align the subspaces corresponding to different types. Since our main aim is to improve the entity embeddings, rather than predicting relationships between entities of different types, methods such as TransH and TransR, which rely on projecting the entities to a different space, are not directly suitable. On the other hand, TransE is only suitable for one-to-one relations.

We propose an alternative to TransE which is inspired by our modeling of semantic types. As in TransE, we assume that every relation $k$ is represented as a vector $r_k$. We furthermore write $\mathit{rhs}(e,k) = \{f \mid (e,k,f) \in G\}$ and $\mathit{lhs}(k,f) = \{e \mid (e,k,f) \in G\}$. Rather than imposing that $e + r_k = f$ if $(e,k,f) \in G$, as in TransE, we require that the points in $P_{e,k} = \{p_f \mid f \in \mathit{rhs}(e,k)\} \cup \{p_e + r_k\}$ lie in a low-dimensional subspace and, similarly, that the points in $P_{k,f} = \{p_e \mid e \in \mathit{lhs}(k,f)\} \cup \{p_f - r_k\}$ lie in a low-dimensional subspace. Note that in the case of one-to-many or many-to-one relations, this part of the model is similar to TransH in the special case where the considered subspaces are one-dimensional. Note that in the case of a one-to-many or many-to-many relation, the set of entities $\mathit{rhs}(e,k)$ is essentially treated as an additional semantic type (e.g. the set of all films directed by Stephen Spielberg), and similar for many-to-one relations and the set $\mathit{lhs}(k,f)$. As for the semantic types we will consider a number of variants:

$$\begin{aligned}
J^{\mathit{dim}}_{\mathit{rel}} &= \sum_{k \in R} \sum_{p \in P_{e,k}} \|p - \sum_{j=0}^{n} \mu^{e,k}_j q^{e,k}_j\|^2 \\
&\quad + \sum_{p \in P_{k,f}} \|p - \sum_{j=0}^{n} \mu^{k,f}_j q^{k,f}_j\|^2 \\
J^{\mathit{dist}}_{\mathit{rel}} &= \sum_{f \in \mathit{rhs}(e,k)} d(p_f, p_e + r_k)^2 + \sum_{e \in \mathit{rhs}(k,f)} d(p_e, p_f - r_k)^2 \\
J_{\mathit{rel}} &= J_{\mathit{rel}} + J^{\mathit{dist}}_{\mathit{rel}}
\end{aligned}$$

where we write e.g. $p \in P_{e,k}$ to sum over all entities $e$ and all points $p$ in $P_{e,k}$. Note that the variant $J^{\mathit{dist}}_{\mathit{rel}}$ essentially corresponds to TransE. For the variants $J_{\mathit{rel}}$ and $J^{\mathit{comb}}_{\mathit{rel}}$ we again use nuclear norm regularization to enforce low-dimensional subspaces. Let the $i^{th}$ row vector of the matrix $M_{e,k}$ be given by $q^{e,k}_i - q^{e,k}_0$ and similar for $M_{k,f}$. We define:

$$J^2_{\mathit{reg}} = \sum_{k \in R} \|M_{e,k}\|_* + \|M_{k,f}\|_*$$

Note that we only need to consider the combination $(e,k)$ or the combination $(k,f)$ if there is at least one triple of the form $(e,k,f)$ in $G$, since otherwise we can trivially choose $M_{e,k}$ and $M_{e,k}$ as the zero matrix. The full regularization term is given by $J_{\mathit{reg}} = J^1_{\mathit{reg}} + J^2_{\mathit{reg}}$.

## 4 EVALUATION

### 4.1 Data acquisition

In our experiments, we have used Wikidata to obtain a set of entities $E$ and their corresponding semantic types. To generate the bag-of-words representation $W_e$ of a given entity, we take advantage of the fact that Wikidata entities $e$ are linked to their corresponding Wikipedia article $d_e$. The set $W_e$ contains the words occurring in $d_e$, as well as the $m$ words before and after any mentions of the entity in other Wikipedia articles. Following [24], we have used a window size of $m = 10$ (but without crossing sentence boundaries). In particular, we treat every link from some Wikipedia article $d_x$ to $d_e$ as a mention of $e$, as well as any repeated occurrences of the corresponding anchor text in $d_x$. The word-word co-occurrence in the $J_{glove}$ component of our model has been obtained from the entire Wikipedia corpus, as in the standard GloVe model. Using the Wikidata dump from October 26, 2015 and the Wikipedia dump from November 02, 2015, we have then selected those entities $e$ which are mentioned in at least 10 Wikipedia articles, resulting in a set $E$ containing 1,292,702 entities. For each semantic type $s$, the set $E_s$ contains those entities which are asserted to be of type $s$ via the *instance of* property as well as all instances which are asserted to belong to one of the supertypes of $s$, which was determined using the *subclass of* property. As the set of binary relations $R$ we considered all Wikidata properties whose value is another entity, apart from *instance of* and *subclass of* which have already been used to determine the sets $E_s$. In the case of Wikipedia, we adopted a fairly straightforward preprocessing strategy, as used in many other works such as [31]. In particular, we removed punctuations, lower-cased the tokens, and conducted sentence segmentation using the NLTK library[8]. We also removed words whose term frequency in the entire collection was less than 10. A script has been made available online[9], which generates an exact copy of our data set, starting from the publicly available dumps of Wikipedia and Wikidata. The implementation of all variants of our model has also been made available at the same link.

Most knowledge graph embedding models have been evaluated on fragments of Freebase and WordNet. Our choice of Wikidata is motivated by the fact that it has relatively clean semantic type information. For example, while Barack Obama is of type *Human* on Wikidata, Freebase among others mentions the following types: *film subject*, *musical artist* and *building occupant*. Furthermore, while Freebase contains information about tens of millions of entities, the standard benchmark datasets, called FB15k [2] and FB13 [28], are relatively small: FB15k covers 14,951 entities and 1,345 relation types, while FB13 covers 74,043 entities and 13 relation types. For completeness, we will include a comparison of our model on these standard benchmark sets for link prediction and triple classification, which are the two standard evaluation tasks for knowledge graph embedding. Our other experiments will be oriented more towards evaluating the usefulness of our model for learning conceptual space representations, in particular their ability to capture semantic relations between entities of the same type (which are not covered in the knowledge graph). This requires a sufficient number of entities for each of the considered semantic types, and a sufficiently clean semantic type structure. Accordingly these tasks will be evaluated only on the WikiData fragment described above. Finally, WordNet has a rich semantic type hierarchy, but contains relatively few instances of these types (e.g. of the 51K leaf nodes in WordNet 1.7 only 7K were found to be instances in [1]) and is thus not suitable for our purposes.

The semantic types of the entities occurring in FB15k and FB13 have been obtained from the "type/instance" field in the Freebase dump[10]. To link Freebase entities to Wikipedia, we have made use of existing Freebase-WikiData mappings[11].

## 4.2 Variants and baseline methods

| name | type | relation | regularization |
|---|---|---|---|
| EECS$_{\text{full}}$ | $J_{type}$ | $J_{rel}$ | $J^1_{reg} + J^2_{reg}$ |
| EECS$_{\text{no rel}}$ | $J_{type}$ | - | $J^1_{reg}$ |
| EECS$_{\text{no type}}$ | - | $J_{rel}$ | $J^2_{reg}$ |
| EECS$_{\text{no NN}}$ | - | $J_{rel}$ | - |
| EECS$_{\text{text}}$ | - | - | - |
| EECS$_{\text{rel-dim}}$ | $J_{type}$ | $J^{dim}_{rel}$ | $J^1_{reg} + J^2_{reg}$ |
| EECS$_{\text{rel-dist}}$ | $J_{type}$ | $J^{dist}_{rel}$ | $J^1_{reg}$ |
| EECS$_{\text{type-comb}}$ | $J^{comb}_{type}$ | $J_{rel}$ | $J^1_{reg} + J^2_{reg}$ |
| EECS$_{\text{type-dist}}$ | $J^{comb}_{type}$ | $J_{rel}$ | $J^2_{reg}$ |

**Table 1:** Overview of considered variants of our model.

Our main baseline is pTransE, which also learns an embedding of entities by combining a word embedding model with a knowledge graph embedding model. We used the it's publicly available implementation[12]. We consider three variants of this baseline: pTransE$_{\text{anch}}$ is the version proposed in [31], which uses anchor text for aligning word vectors and entity vectors; pTransE$_{\text{art}}$ is the improvement proposed in [35], which uses the words in the Wikipedia article $d_e$ instead of anchor text (and a slightly different model); pTransE$_{\text{full}}$ is a variant of pTransE$_{\text{art}}$, which uses the bag of words representation $W_e$ instead, as in our method. In addition, we compare our method against RESCAL, as well as a number of knowledge graph embedding methods: TransE, TransH, TransR and CTransR. The source codes of these translation-based models are publicly available online[13]. RESCAL [23] is a collective matrix factorization model based on tensor factorization, which has been designed to account for the inherent structure of dyadic relational data. The implementation of RESCAL can be found here[14]. For the knowledge graph embedding methods, we used Bernoulli sampling for selecting negative examples (see [31]); we also obtained results for uniform sampling (not shown), and found the results to be very similar to Bernoulli sampling but slightly worse. It is expected that all of these methods will perform worse than both pTransE and our model, as they cannot exploit the text representation $W_e$ of the entities. We also compare our method with Skip-gram and CBOW, which can only use text representations and are thus also expected to perform worse. In particular, to apply these models to learn entity embeddings, we use the same method as for our model to determine entity mentions on Wikipedia, and then apply the standard models based on the words surrounding these mentions. Finally, we have compared our method with the multi-dimensional scaling (MDS) based method from [6], in which case we learn a separate vector space for every semantic type. Because of the limited scalability of the latter model, however, we have only considered this for semantic types with up to 10000 instances. Following [6], for each of the remaining semantic types, a vector space representation of the corresponding entities was obtained using Positive Pointwise Mutual Information (PPMI). We then applied multi-dimensional scaling to obtain a lower-dimensional representation, using the angular difference between the initial vectors as metric. We have used the MDS model implemented in MATLAB. We have also considered the method from [14] as an additional baseline, but found that this method could not scale to even the reduced data set that we used for the MDS experiments.

Throughout this section, we will refer to our model as EECS (Entity Embeddings with Conceptual Subspaces). As an ablation study, we will consider a number of variants of our model in which some components have been removed. EECS$_{\text{full}}$ refers to our full model, in which $J_{type}$ is used for modelling semantic types and $J_{rel}$ is used for modelling relations; EECS$_{\text{no rel}}$ refers to a variant in which $J_{rel}$

[8] http://www.nltk.org/

[9] https://github.com/bashthebuilder/ECAI-2016/blob/master/README.md

[10] https://developers.google.com/freebase/data

[11] https://developers.google.com/freebase/data#freebase-wikidata-mappings

[12] https://github.com/Mrlyk423/Relation_Extraction

[13] https://github.com/Mrlyk423/Relation_Extraction

[14] https://github.com/mnick/rescal.py

and the associated regularization component $J^2_{reg}$ have been removed; $\text{EECS}_{\text{no type}}$ refers to a variant in which $J_{type}$ and $J^1_{reg}$ have been removed; $\text{EECS}_{\text{no NN}}$ refers to a variant in which the regularization component $J_{reg}$ has been removed (which also trivializes the component $J_{type}$); $\text{EECS}_{\text{text}}$ refers to a variant in which only the component $J_{text}$ is used, reducing our model essentially to a variant of GloVe.

Furthermore, we have considered a few variants of $\text{EECS}_{\text{full}}$ in which we change the component $J_{type}$ or $J_{rel}$ by one of the proposed alternatives: $\text{EECS}_{\text{rel-dim}}$ refers to a variant in which $J_{rel}$ is replaced by $J^{dim}_{rel}$, $\text{EECS}_{\text{rel-dist}}$ refers to a variant in which $J_{rel}$ is replaced by $J^{dist}_{rel}$, $\text{EECS}_{\text{type-comb}}$ refers to a variant in which $J_{type}$ has been replaced by $J^{comb}_{type}$, and $\text{EECS}_{\text{type-dist}}$ refers to a variant in which only distance information is considered for modelling semantic types (which corresponds to using $J^{comb}_{type}$ without regularization). An overview of the considered variants of our model is provided in Table 1.

## 4.3 Methodology

All experiments were evaluated using five-fold cross validation. For tuning the parameter $\beta$ of our model, based on a tuning/validation set in each experiment, we considered the range $\{50, 100, 150, 200, 250, 300, 350, 400\}$. For the parameter $\alpha$, we considered values between 0 and 1 with an increment of 0.1. The number of iterations for all models was set to 20, as we found that beyond this number empirical results became fairly consistent in all cases. Based on the tuning set, in each of the experiments the optimal value of $\beta$ was found to be 300, while the optimal values of $\alpha$ varied between 0.4 and 0.7. The number of dimensions was always set to 300 for our model, noting that because of the nuclear norm regularization this only represents an upper bound on the actual number of dimensions. All parameters of the baseline methods, including the number of dimensions, have been optimized based on the tuning set in each experiment. For the MDS method, the number of dimensions was tuned for each semantic type separately (as this method learns a separate vector space for each semantic type), considering the range from 10 to 100 in steps of 10. For the remaining baselines, which construct a single vector space, the number of dimensions was varied between 50 to 300 in steps of 50.

Our model has been implemented in C using standard POSIX threads, which helps scale our implementation to large text collections. For example, for the considered 1.2 million Wikidata entities our full model takes about 30 minutes per iteration using 8 threads, scaling almost linearly in the number of entities. In contrast, $\text{EECS}_{\text{text}}$ takes about 18 minutes for each iteration using 8 threads.

## 4.4 Results

We will evaluate our model on four different tasks: *ranking*, *induction*, *analogy making*, and *knowledge graph embedding*. The first two of these tasks are directly aimed at evaluating to what extent the type-specific subspaces learned by our model are useful as conceptual space representations. In particular, *ranking* will evaluate to what extent important features of a given semantic type can indeed be modelled as directions in the associated subspace, while *induction* assesses to what extent we can use these representations to find new instances of a given concept, given only a few example instances. The *analogy making* task is aimed at evaluating how well the different subspaces are aligned. As discussed above, these first three tasks will be evaluated using a large fragment of WikiData. The motivation behind the fourth tasks relates to the observation that even though our motivation was rather different from the motivation behind pTransE, both our model and pTransE combine a word based entity embedding component with a knowledge graph embedding component. Since pTransE has proven a successful approach for knowledge graph embedding, we want to analyse whether our model has any advantages in such a setting. As explained above, for this task we will use the standard benchmark datasets FB15k and FB13.

| Semantic Type | Number of Entities | NN-Dimensions |
|---|---|---|
| human | 191211 | 288 |
| railway station | 4120 | 121 |
| house | 2762 | 136 |
| organization | 1379 | 88 |
| national park | 1307 | 56 |
| building | 1269 | 52 |
| food | 1155 | 55 |
| college | 858 | 33 |
| automobile | 31 | 12 |
| candy | 10 | 2 |

**Table 2:** Number of dimensions selected by the nuclear norm (NN) regularization component of our model for some of the semantic types.

| | Ranking | Induction | | | Analogy |
|---|---|---|---|---|---|
| | $\rho$ | MAP | P@5 | MRR | Acc. |
| Skip-gram | 0.155 | 0.176 | 0.356 | 0.505 | 0.184 |
| CBOW | 0.159 | 0.182 | 0.350 | 0.500 | 0.213 |
| RESCAL | 0.081 | 0.020 | 0.189 | 0.423 | 0.371 |
| TransE | 0.110 | 0.060 | 0.200 | 0.451 | 0.382 |
| TransH | 0.142 | 0.072 | 0.210 | 0.415 | 0.382 |
| TransR | 0.100 | 0.102 | 0.302 | 0.489 | 0.378 |
| CTransR | 0.122 | 0.132 | 0.323 | 0.499 | 0.402 |
| $\text{pTransE}_{\text{anch}}$ | 0.099 | 0.101 | 0.301 | 0.488 | 0.476 |
| $\text{pTransE}_{\text{art}}$ | 0.202 | 0.218 | 0.475 | 0.751 | 0.512 |
| $\text{pTransE}_{\text{full}}$ | 0.213 | 0.224 | 0.490 | 0.756 | 0.532 |
| $\text{EECS}_{\text{full}}$ | **0.319** | **0.231** | **0.609** | **0.883** | 0.591 |
| $\text{EECS}_{\text{no rel}}$ | 0.301 | 0.229 | 0.588 | 0.868 | 0.552 |
| $\text{EECS}_{\text{no type}}$ | 0.266 | 0.225 | 0.585 | 0.854 | 0.549 |
| $\text{EECS}_{\text{no NN}}$ | 0.258 | 0.220 | 0.581 | 0.843 | 0.545 |
| $\text{EECS}_{\text{text}}$ | 0.254 | 0.218 | 0.579 | 0.831 | 0.540 |
| $\text{EECS}_{\text{type-comb}}$ | 0.312 | **0.231** | 0.601 | **0.883** | **0.595** |
| $\text{EECS}_{\text{type-dist}}$ | 0.295 | **0.231** | 0.585 | 0.858 | 0.550 |
| $\text{EECS}_{\text{rel-dim}}$ | 0.309 | 0.225 | 0.585 | 0.859 | 0.551 |
| $\text{EECS}_{\text{rel-dist}}$ | 0.299 | 0.225 | 0.585 | 0.855 | 0.549 |

**Table 3:** Experimental results for the full WikiData test data.

| | Ranking | Induction | | | Analogy |
|---|---|---|---|---|---|
| | $\rho$ | MAP | P@5 | MRR | Acc. |
| MDS | 0.101 | 0.121 | 0.231 | 0.388 | 0.354 |
| $\text{EECS}_{\text{full}}$ | 0.218 | 0.140 | 0.301 | 0.463 | 0.456 |

**Table 4:** Comparison with MDS on a subset of the WikiData test data.

An important aspect in the discussion of the result is to analyze the effectiveness of nuclear norm regularization in identifying the most appropriate number of dimensions for each of the semantic types. To illustrate the behaviour of this regularization component, Table 2 shows the number of dimensions that was found for a few notable semantic types (when using the default configuration of our model). As expected, semantic types with more entities generally end up being associated with higher-dimensional subspaces, but other factors affect the choice as well. For example, note that *house* has fewer instances than *railway station*, while being represented by a higher-dimensional subspace. Intuitively, this reflects the idea that the type *house* is more diverse or complex than the type *railway station*.

### 4.4.1 Ranking

A characteristic feature of conceptual spaces is that they are encoded as Cartesian products of interpretable dimensions. For a vector space model to be meaningful as a conceptual space, it is therefore important that salient properties can be modelled as directions[15]. Therefore, we have evaluated the ability of our model to correctly rank entities according to a given property. As we need the ground truth, we have focused on properties with numerical values which are available in Wikidata (but have not been considered when learning the space), e.g. the *date of birth* for entities of type *human*, or the *boiling point* of entities of type *chemical element*. In total, we have retrieved 26 numerical attributes which are available for at least 30 entities. Some of these numerical attributes appear for different semantic types (e.g. the property *inception*, referring to the foundation year, applies to the semantic types *film*, *organization* and *country*, among others). In total, we obtained 73 such property-type combinations, by considering for each numerical attribute all the maximally specific semantic types with at least 30 instances that have the attribute. Each of these 73 combinations was considered as a problem instance. For each problem instance, the corresponding set of entities is split into 60% training, 20% validation and 20% testing sets. The full specification of the 73 problem instances and corresponding splits is available online[16]. From the training set, a direction is estimated using the SVMRank model[17] [18]. The parameters of the resulting ranking models are optimized using the validation sets. Table 3 shows the performance on the testing set, in terms of Spearman's $\rho$[18], expressing the correlation between the ranking predicted by the model and the ranking according to the numerical values found in Wikidata.

The results show that standard word and knowledge graph embedding models are not competitive, which is not surprising given that they use less information than our model. However, the results also show that our model substantially outperforms pTransE, even the variant pTransE$_{\text{full}}$ which uses the same input as our model. Comparing the results for the variants of our model, we notice that the relation component only has a small impact, i.e. the performance of EECS$_{\text{no rel}}$ is close to EECS$_{\text{full}}$, and similarly, the performance of EECS$_{\text{no type}}$ is close to EECS$_{\text{text}}$. Regarding the variants of $J_{type}$, EECS$_{\text{full}}$ and EECS$_{\text{type-comb}}$ are clearly better than EECS$_{\text{type-dist}}$, which shows the importance of nuclear norm regularization for identifying low-dimensional subspaces. The results in Table 4 compare our model against the MDS model from [6] on a reduced set of 27 problem instances. These results clearly show that the MDS method is not competitive.

Tables 5 and 6 illustrate the results of the ranking experiment for three attributes, by showing the 5 lowest and 5 highest ranked entities respectively. Note that Table 5 starts with the lowest ranked entity (i.e. the entity that has the lowest value for the considered attribute), while Table 6 starts with the highest ranked entity (i.e. the entity that has the highest value for the considered attribute). While the rankings are not perfect (e.g. Bermuda, Monaco, San Marino and Barbados are all less populous than Malta, Ptolemy lived around 500 years after Plato), the model's ability to separate high-scoring entities from low-scoring entities is nonetheless remarkable, considering that none of the information that was used to learn the vector space explicitly referred to these attributes.

| Population | Inception | Date of Birth |
|---|---|---|
| Malta | General Electric | Valmiki |
| Bermuda | IBM | Jesus Christ |
| Monaco | Hewlett Packard | Cleopatra |
| San Marino | Microsoft | Ptolemy |
| Barbados | Oracle Corporation | Plato |

**Table 5:** Five lowest ranked entities for a number of ranking problem instances.

| Population | Inception | Date of Birth |
|---|---|---|
| China | Alphabet Inc. | Prince George of Cambridge |
| India | Tencent Holdings | Isabela Moner |
| USA | Facebook, Inc. | Justin Bieber |
| Soviet Union | Uber | Lionel Messi |
| Brazil | Amazon.com | Kim Kardashian |

**Table 6:** Five highest ranked entities for a number of ranking problem instances.

### 4.4.2 Induction

A second characteristic feature of conceptual spaces is that properties correspond to convex regions. Moreover, it is often assumed that the boundaries of these regions are determined based on the distance to a particular point in the space, which acts as a prototype. In this experiment, we test our method's ability to make inductive inferences based on this view. In particular, given a number of entities of the same type which have some property in common, the task we consider is to identify other entities that also have this property (without any knowledge about the property being considered).

Problem instances in this case were obtained by omitting all triples of the form $(., r, f)$ for particular choices of $r$ and $f$, when learning the embeddings. The set of entities $e$ for which $(e, r, f) \in G$ then defines a problem instance. For example, the property being considered could be "films directed by Stephen Spielberg". Given a few examples of such films, the task is to identify other films directed by Stephen Spielberg (but without the knowledge that this is the property being considered). For each $(r, f)$ combination, the set of entities $\{e \mid (e, r, f) \in G\}$ is split into 60% training, 20% tuning and 20% testing sets. Details on the $(r, f)$ combinations and associated splits are available online[19].

For evaluation purposes, we consider this task as a ranking task. In particular, for each problem instance, we rank the entities of the associated semantic type (defined as the most specific semantic type that contains all the considered entities) based on their distance to the center-of-gravity of the training instances, and evaluate the quality of this ranking using mean average precision (MAP), Precision@5 (P@5) and Mean Reciprocal Rank (MRR); note that in all cases, higher values are better.

The results in Table 3 show that our model again substantially outperforms all of the baselines. Note, however, that in the case of MAP, the differences with pTransE$_{\text{full}}$ are rather small. The fact that the differences are clearer for P@5 and MRR suggests that our method is better able to select a few entities with high precision. The MAP score tends to be dominated by outliers, leading to smaller differences. Regarding the different variants of our model, the semantic type component and relation component now play a more equal role, given the rather similar performance of EECS$_{\text{no rel}}$ and

[15] Conceptual space representations also encode information about the correlation between the underlying dimensions, which in our case is captured by the angles between these directions.

[16] https://github.com/bashthebuilder/ECAI-2016

[17] `https://www.cs.cornell.edu/people/tj/svm_light/svm_rank.html`

[18] The reported average $\rho$ values have been obtained using the Fisher z-transformation.

[19] `https://github.com/bashthebuilder/ECAI-2016`

EECS$_{\text{no type}}$, although semantic type information is still more important than the knowledge graph information (as EECS$_{\text{no rel}}$ performs better than EECS$_{\text{no type}}$). As for the ranking experiment, we notice that using $J_{type}^{dist}$ in EECS$_{\text{no type}}$ leads to worse results, highlighting again the importance of nuclear norm regularization. A before, the MDS model is not competitive.

### 4.4.3 *Analogy making*

Finally we have considered the problem of completing analogical proportions of the form "$a$ is to $b$ what $c$ is to ...", which is a standard evaluation task for word embeddings. Our main aim in this task is to evaluate how well different subspaces are aligned. We have used the test sets from the GloVe project[20] that are about entities, resulting in a total of 8363 problem instances. As there is no need for training data, in this case we randomly split the data into 25% tuning and 75% testing sets. Full details on the test sets and splits that were used have been made available online[21].

The results are largely consistent with the findings from the previous two experiments. The main difference is that the variant EECS$_{\text{type-comb}}$ slightly outperforms EECS$_{\text{full}}$ in this case. The rather large difference in performance between EECS$_{\text{type-comb}}$ and EECS$_{\text{type-dist}}$ again clearly illustrates the impact of nuclear norm regularization on the results.

### 4.4.4 *Knowledge graph embedding*

We have also conducted two knowledge graph embedding experiments using the benchmark datasets FB15k and FB13. In particular, we have evaluated our method on the widely used Link Prediction and Triple Classification tasks.

**Link prediction** For the link prediction task [2], given an entity $e$ and a relation $r$, the aim is either to find an entity $f$ such that $(e, r, f)$ or to find an entity $f$ such that $(f, r, e)$. We have used the standard FB15k test set for this evaluation, which allows us to compare our model with the published results of the state-of-the-art knowledge graph embedding models. Two widely used evaluation metrics, which we will also use, are the average rank of correct entities, called "Mean Rank", and "HITS@10", which is defined as the proportion of test triples in which the target entity was ranked in the top 10. Note that the Mean Rank score is to be minimized while the HITS@10 score is to be maximized. We have used the standard evaluation protocol, including the Bernoulli sampling trick to corrupt the head or tail entity. Test instances were not filtered (which corresponds to the so-called raw version of the task).

In Table 7 we show that our model clearly outperforms the standard baselines in both metrics. To a large extent, this is due to the fact that, apart from pTransE, the other models do not exploit the bag-of-words representations of the entities. Note that we do not show results for EECS$_{\text{text}}$ and EECS$_{\text{no rel}}$ in the tables because these models do not take into account any input from the knowledge graph, and is therefore not suitable. The baselines Skip-gram and CBOW are not considered for the same reason.

**Triple classification** The objective in triple classification [28] is to judge whether a given triplet $(e, r, f)$ is correct or not, i.e. whether entities $e$ and $f$ are in relation $r$ with each other. This can be naturally cast as a binary classification task. We present results on FB13 and FB15k. The FB13 dataset already comes with golden negative triplets, while we followed the methodology from [28] to construct negative samples for FB15k. For the classification task, we need to set a threshold $\delta_r$ for each relation $r$. We obtain $\delta_r$ by maximizing the classification accuracies on the validation set. For the given triplet, if the energy score is larger than the relation-specific $\delta_r$, the instance will be classified as positive, otherwise negative. This is the standard experimental setting for this evaluation task.

| Models | Link Prediction (FB15k) | | Triple Classification | |
|---|---|---|---|---|
| | Mean Rank | HITS@10 | FB13 | FB15k |
| RESCAL | 683 | 44.1 | 65.3 | 71.6 |
| TransE | 125 | 47.1 | 81.5 | 79.8 |
| TransH | 87 | 64.4 | **83.3** | 79.9 |
| TransR | 77 | 68.7 | 82.5 | 82.1 |
| CTransR | 75 | 70.2 | - | 84.3 |
| pTransE$_{\text{anch}}$ | 58 | 84.6 | 73.3 | 74.3 |
| pTransE$_{\text{art}}$ | 55 | 85.3 | 75.8 | 75.5 |
| pTransE$_{\text{full}}$ | 51 | 86.4 | 76.3 | 77.4 |
| EECS$_{\text{full}}$ | 48 | 89.7 | 83.1 | 89.6 |
| EECS$_{\text{no type}}$ | 56 | 84.7 | 71.2 | 82.1 |
| EECS$_{\text{no NN}}$ | 59 | 82.7 | 70.1 | 81.4 |
| EECS$_{\text{type-comb}}$ | **47** | **89.9** | **83.3** | **89.9** |
| EECS$_{\text{type-dist}}$ | 54 | 83.2 | 81.1 | 82.1 |
| EECS$_{\text{rel-dim}}$ | 54 | 85.1 | 79.3 | 88.2 |
| EECS$_{\text{rel-dist}}$ | 52 | 85.3 | 78.8 | 87.1 |

**Table 7:** Link prediction and Triple classification results.

Our experimental results are shown in Table 7. With the exception of the pTransE variants, we again show the published results for the baseline models. The baseline results have been reported in [33, 15, 17, 22]. On the FB13 dataset, our model matches the performance of the TransH model, although we clearly outperform all baselines on the FB15k dataset. This means that on the FB13 dataset, the bag-of-words representations of the entities cannot be exploited effectively, although our model is still not at a disadvantage. This seems related to the fact that only 13 relation types are considered in FB13, which were moreover specifically selected such that they can be predicted from each other, in the sense that hard-to-predict relation types have been removed [28].

## 5 CONCLUSIONS

We have proposed a new method for learning vector-space embeddings of entities, based on available semantic information (from Wikidata) and textual descriptions (from Wikipedia). From a technical point of view, the main novelty of our model is the use of nuclear norm regularization to encode the requirement that entities of the same semantic type should lie in a lower-dimensional subspace. In particular, nuclear norm regularization allows the model to automatically select the most appropriate number of dimensions for the subspace corresponding to each type. From an application point of view, our main motivation was to learn subspaces that are useful as approximations of conceptual spaces. To support this view, among others, we have shown that many numerical attributes can be faithfully modelled as directions and that the learned representations allow us to model induction based on distance to a centroid. In addition, we have also obtained good results for analogy making and for two standard knowledge graph embedding tasks.

## 6 ACKNOWLEDGEMENTS

This work was supported by ERC Starting Grant 637277.

[20] `http://nlp.stanford.edu/projects/glove/`

[21] `https://github.com/bashthebuilder/ECAI-2016`

# REFERENCES

[1] Enrique Alfonseca and Suresh Manandhar, 'Distinguishing concepts and instances in wordnet', in *Proceedings of the First International Conference of the Global WordNet Association*, (2002).

[2] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, 'Translating embeddings for modeling multi-relational data', in *Proceedings of the Annual Conference on Neural Information Processing Systems*, 2787–2795, (2013).

[3] A. Bordes, J. Weston, R. Collobert, and Y. Bengio, 'Learning structured embeddings of knowledge bases', in *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, (2011).

[4] Antonio Chella, 'A cognitive architecture for music perception exploiting conceptual spaces', in *Applications of Conceptual Spaces*, eds., Frank Zenker and Peter Grdenfors, 187–203, Springer International Publishing, (2015).

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

[6] J. Derrac and S. Schockaert, 'Inducing semantic relations from conceptual spaces: a data-driven approach to plausible reasoning', *Artificial Intelligence*, 74–105, (2015).

[7] X. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy, T. Strohmann, S. Sun, and W. Zhang, 'Knowledge vault: A web-scale approach to probabilistic knowledge fusion', in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 601–610, (2014).

[8] I. Douven, L. Decock, R. Dietz, and P. Égré, 'Vagueness: A conceptual spaces approach', *Journal of Philosophical Logic*, **42**, 137–160, (2013).

[9] Katrin Erk, 'Representing words as regions in vector space', in *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, pp. 57–65, (2009).

[10] Maryam Fazel, *Matrix rank minimization with applications*, Ph.D. dissertation, Stanford University, 2002.

[11] Leon Festinger, *A theory of cognitive dissonance*, Stanford University Press, 1957.

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

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

[14] S. Guo, Q. Wang, B. Wang, L. Wang, and L. Guo, 'Semantically smooth knowledge graph embedding', in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing*, pp. 84–94, (2015).

[15] Shizhu He, Kang Liu, Guoliang Ji, and Jun Zhao, 'Learning to represent knowledge graphs with gaussian embedding', in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pp. 623–632. ACM, (2015).

[16] Cho-Jui Hsieh and Peder Olsen, 'Nuclear norm minimization via active subspace selection', in *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pp. 575–583, (2014).

[17] Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao, 'Knowledge graph embedding via dynamic mapping matrix', in *Proceedings of ACL*, pp. 687–696, (2015).

[18] Thorsten Joachims, 'Optimizing search engines using clickthrough data', in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 133–142. ACM, (2002).

[19] Antonio Lieto, Andrea Minieri, Alberto Piana, and Daniele P. Radicioni, 'A knowledge-based system for prototypical reasoning', *Connection Science*, **27**, 137–152, (2015).

[20] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu, 'Learning entity and relation embeddings for knowledge graph completion', in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pp. 2181–2187, (2015).

[21] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean, 'Distributed representations of words and phrases and their compositionality', in *Proceedings of the 27th Annual Conference on Neural Information Processing Systems*, pp. 3111–3119, (2013).

[22] Dat Quoc Nguyen, Kairit Sirts, Lizhen Qu, and Mark Johnson, 'STransE: a novel embedding model of entities and relationships in knowledge bases', in *Proceedings of the 15th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, to appear*, (2016).

[23] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel, 'A three-way model for collective learning on multi-relational data', in *Proceedings of the 28th International Conference on Machine Learning*, pp. 809–816, (2011).

[24] Jeffrey Pennington, Richard Socher, and Christopher D. Manning, 'Glove: Global vectors for word representation', in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pp. 1532–1543, (2014).

[25] Benjamin Recht, Maryam Fazel, and Pablo A Parrilo, 'Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization', *SIAM review*, **52**, 471–501, (2010).

[26] Eleanor H Rosch, 'Natural categories', *Cognitive Psychology*, **4**(3), 328–350, (1973).

[27] Steven Schockaert and Henri Prade, 'Interpolative and extrapolative reasoning in propositional theories using qualitative knowledge about conceptual spaces', *Artificial Intelligence*, **202**, 86–131, (2013).

[28] Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng, 'Reasoning with neural tensor networks for knowledge base completion', in *Advances in Neural Information Processing Systems*, pp. 926–934, (2013).

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

[30] Luke Vilnis and Andrew McCallum, 'Word representations via Gaussian embedding', in *Proceedings of the International Conference on Learning Representations*, (2015).

[31] Z. Wang, J. Zhang, J. Feng, and Z. Chen, 'Knowledge graph and text jointly embedding', in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pp. 1591–1601, (2014).

[32] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen, 'Knowledge graph embedding by translating on hyperplanes', in *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, pp. 1112–1119, (2014).

[33] Han Xiao, Minlie Huang, Yu Hao, and Xiaoyan Zhu, 'TransG: A generative mixture model for knowledge graph embedding', *arXiv preprint arXiv:1509.05488*, (2015).

[34] C. Xu, Y. Bai, J. Bian, B. Gao, G. Wang, X. Liu, and T.-Y. Liu, 'RC-NET: A general framework for incorporating knowledge into word representations', in *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pp. 1219–1228, (2014).

[35] Huaping Zhong, Jianwen Zhang, Zhen Wang, Hai Wan, and Zheng Chen, 'Aligning knowledge and text embeddings by entity descriptions', in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 267–272, (2015).