A Mixture-of-Experts Model for Learning Multi-Facet Entity Embeddings

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

Zied Bouraoui
CRIL - CNRS
Artois University, France
zied.bouraoui@cril.fr

Shelan Jeawak
Department of CS & Creative Technologies
University of the West of England, UK
shelan.jeawak@uwe.ac.uk

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

Abstract

Various methods have already been proposed for learning entity embeddings from text descriptions. Such embeddings are commonly used for inferring properties of entities, for recommendation and entity-oriented search, and for injecting background knowledge into neural architectures, among others. Entity embeddings essentially serve as a compact encoding of a similarity relation, but similarity is an inherently multi-faceted notion. By representing entities as single vectors, existing methods leave it to downstream applications to identify these different facets, and to select the most relevant ones. In this paper, we propose a model that instead learns several vectors for each entity, each of which intuitively captures a different aspect of the considered domain. We use a mixture-of-experts formulation to jointly learn these facet-specific embeddings. The individual entity embeddings are learned using a variant of the GloVe model, which has the advantage that we can easily identify which properties are modelled well in which of the learned embeddings. This is exploited by an associated gating network, which uses pre-trained word vectors to encourage the properties that are modelled by a given embedding to be semantically coherent, i.e. to encourage each of the individual embeddings to capture a meaningful facet.

1 Introduction

Entity embeddings are vector space representations of the entities from a given domain. Such representations are commonly used in cognitive science, where they are referred to as semantic spaces or conceptual spaces (Gärdenfors, 2000). As another example, the field of Information Retrieval also has a long tradition of using vector space representations (Salton, 1973; Deerwester et al., 1990). In the field of Natural Language Processing (NLP), recent years have witnessed an explosion of applications that rely on entity embeddings. For instance, entity embeddings are now commonly used for injecting background knowledge (Logan et al., 2019; Lin et al., 2019), and as core representations for recommender systems (Zhang et al., 2016) and entity-focused search (Van Gysel et al., 2016; Jameel et al., 2017; Zhang et al., 2019). Entity embeddings are learned using a variety of different inputs, ranging from human similarity judgements, to text descriptions, web tables and images. Regardless of how they are learned, entity embeddings can essentially be viewed as compact encodings of a similarity relation. Indeed, while many embeddings exhibit various interesting linear regularities, such regularities are the result of the structure of the similarity relation that is used for learning the embedding (Allen and Hospedales, 2019).

Similarity is inherently multi-faceted, with the importance of different facets being context dependent. For instance, two movies can be similar because they belong to the same genre or because they are about the same historic event, among many others. However, these different facets of similarity are not reflected in the structure of standard entity embeddings. To see why this is sub-optimal, consider the problem of concept induction: given a small set of entities $e_1, ..., e_k$, identify other entities that are of the same kind. For instance, given the examples *Barcelona*, *Madrid*, *Alicante*, valid completions would be other Spanish cities. The problem of concept induction underpins many of the applications in which entity embeddings are used, including knowledge base completion and recommendation. In cases where the given set of entities is small, the result will strongly depend on the similarity relation encoded by the given entity embedding: given a few entities, we can do little else than selecting the nearest neighbors of their (averaged) entity vectors. However, if we have several embeddings of the considered entities, each capturing different facets, then we can solve the concept induction task by first identifying the most relevant facet(s), and thus rely on a form of similarity that is relevant for the given concept.

The problem of learning separate facet-specific embeddings is related to disentangled representation learning. While this latter problem has already received considerable attention, most existing work has focused on (semi)-supervised settings, primarily in the visual domain. Unsupervised approaches for disentangled representation learning generally need strong inductive biases (Locatello et al., 2019). In the text domain, most work has focused on separating style or sentiment from content. However, one recent exception is (Alshaikh et al., 2019), where an iterative method is proposed to decompose a given entity embedding into facet-specific vector spaces. To provide the required inductive bias, they first determine which properties are captured by the given entity embedding. These properties correspond to words from text descriptions of the entities, whose occurrence can be predicted from the entity vectors. To identify words that are likely to describe properties from the same facet, they rely on the intuition that such properties should have similar word vectors, in a given pre-trained word embedding.

The experimental results from Alshaikh et al. (2019) show that learning facet-specific embeddings is indeed helpful for concept induction. However, their method is applied to entity embeddings that have been learned from text descriptions using multi-dimensional scaling (MDS), which has two important limitations. First, MDS has a quadratic space complexity, which makes it unsuitable for large domains. Second, and most fundamentally, they crucially rely on the assumption that facets of interest correspond to linear sub-spaces of the initial entity embedding. As another limitation of their method, the assumption that facets can be identified with clusters in a word embedding space seems too strong. While words that describe properties of the same kind (e.g. different names of movie genres) are indeed often clustered together in a word embedding, the range of words that are relevant to a given facet is usually more varied (e.g. adjectives such as *scary* are relevant when modelling genre, but this word may not be clustered together with genre names). To address these issues, we propose a method that directly learns multi-facet entity embeddings from text descriptions. To this end, we use a mixture-of-experts formulation (Jacobs et al., 1991), in which the experts essentially correspond to GloVe models (Pennington et al., 2014), each focusing on a subset of the vocabulary. The decision on which words are modelled by which experts is made by a so-called gating network, which uses pre-trained word vectors as input. In this way, we can capture the intuition that words which are relevant to the same facet typically have similar word embedding representations, without having to assume that all such word appear in a single cluster.

2 Related Work

Conceptual spaces. The idea that a single vector space is insufficient for modelling similarity has been widely studied in cognitive science. In particular, this idea is closely related to the distinction between so-called *integral* and *separable* dimensions, which plays a central role in cognitive models of categorisation (Gärdenfors, 2000). Dimensions, in this context, refer to elementary cognitive features. Two dimensions are intuitively separable if they can be considered in isolation (e.g. *size* and *hue*), and integral otherwise (e.g. *hue*, *saturation*, and *luminosity* are jointly perceived as colour). Psychological studies have shown that the way in which humans generalize from examples is affected by the nature of the underlying dimensions (Grau and Nelson, 1988; Nosofsky and Palmeri, 1996). The theory of conceptual spaces (Gärdenfors, 2000) is a popular cognitive model which takes this distinction between integral and separable dimensions into account by organizing dimensions into domains. Dimensions from the same domain are assumed to be integral, whereas those from different domains are assumed to be separable. Each domain is associated with a metric space. Given a conceptual space, the dissimilarity between two objects is determined by (i) computing their (usually Euclidean) distance in each of the domain-specific spaces and (ii) taking a weighted average of these distances. This is in accordance with empirical findings, which suggest that Euclidean distance is predictive of human similarity judgements in the case of integral dimensions, whereas such judgements are a function of Manhattan distance in the case of separable dimensions; we refer to (Gärdenfors, 2000) for more details.

The problem of learning conceptual spaces from data has only received limited attention. Inspired by conceptual spaces, Derrac and Schockaert (2015) introduced a method for structuring a given entity embedding using interpretable (but non-orthogonal) dimensions. This method was used in the approach by Alshaikh et al. (2019), which consists of (i) identifying interpretable dimensions in the given entity embedding, (ii) clustering the words describing these dimensions, (iii) identifying the linear subspace that best corresponds to the most dominant cluster, (iv) repeating the same method on the orthogonal complement of this subspace. Somewhat related, Rothe and Schütze (2016) propose a supervised method to decompose word embeddings into subspaces that capture particular aspects of word meaning, such as sentiment polarity or part-of-speech. The idea of decomposing a word embedding space into sub-spaces is also central to the method from (Ali et al., 2019), which is aimed at distinguishing synonyms from antonyms. Within a broader context, Banaee et al. (2018) propose a method to group numerical features into domains, with the aim of generating better linguistic descriptions of numerical data.

Disentangled representation learning. The aim of disentangled representation learning is to obtain embeddings, often referred to as latent codes in this context, whose individual dimensions have a clear interpretable meaning. While this is related to our aims in this paper, it should be noted that our focus is on finding sub-spaces that capture different facets of similarity, regardless of whether the individual dimensions are interpretable. Disentangled representation learning has mainly been studied in the context of images, where having a disentangled representation allows one to manipulate images in a given prescribed way (e.g. generating an image showing what a person would look like when wearing glasses). Apart from this particular use case, disentangled representations have also said to lead to more robust models (e.g. being less susceptible to adversarial attacks), and help in transfer learning and few shot learning settings. Existing models mostly correspond to variants of Generative Adversarial Networks, e.g. InfoGAN (Chen et al., 2016), or variational autoencoders, e.g. (Higgins et al., 2017; Kim and Mnih, 2018; Chen et al., 2018). Such models essentially try to find independent factors of variations in the dataset, which is most successful if there is a lot of regularity in the dataset. For instance, a typical application is to learn latent codes of facial images, where factors such as gender, the presence of glasses, or the rotation of the head can be discovered. When learning entity embeddings from text, however, similar strategies tend to be far less successful. In preliminary experiments with InfoGAN, for instance, we were not able to identify any meaningful dimensions for the datasets considered in this paper. In other settings, disentangled representation learning for text has proven more useful. For instance, several authors have focused on separating style (or sentiment) from content (John et al., 2019). In general, most existing approaches for text use some kind of supervision signal, such as aspect-specific similarity judgements (Jain et al., 2018) or sentiment labels (He et al., 2017).

3 Model Description

The main idea underpinning the mixtures-of-experts (MoE) model (Jacobs et al., 1991) is to train a neural network by (i) learning a soft partition of the feature space and (ii) training a separate neural network for each partition class. The individual neural networks, referred to as *experts*, are thus specialized towards the examples from the corresponding partition class. These experts are jointly trained with a so-called *gating network*, which is used to determine the (soft) partition. To apply this model to our setting, we thus need to determine the structure of the gating network and the nature of the experts.

Our aim is to learn facet-specific entity embeddings from the bag-of-words representations (BoW) of a given set of entities. To apply the MoE model to this problem setting, we need an embedding method that can be formulated as a classification or regression problem. Moreover, to allow for an effective gating network, we need the ability to efficiently determine how well different properties are captured by the different entity embeddings. To address both issues, we build on the GloVe word embedding model (Pennington et al., 2014), which is a common choice for learning entity embeddings from BoW representations (Jameel and Schockaert, 2016). Using the notations and terminology of entity embeddings, the GloVe model can be formulated as follows:

$$G = \sum_{j=1}^{m} G_j \qquad G_j = \sum_{i:x_{ij}>0} f(x_{ij}) \left(\mathbf{e_i} \cdot \tilde{\mathbf{w}}_\mathbf{j} + b_i + \tilde{b}_j - \log x_{ij}\right)^2 \qquad f(x_{ij}) = \left(\frac{x_{ij}}{100}\right)^{0.75} \tag{1}$$

Here $\mathbf{e_i}$ represents the embedding of entity e_i, $\tilde{\mathbf{w}}_\mathbf{j}$ is a representation of the word w_j, b_i and $\tilde{b}_j$ are bias terms, x_{ij} is the number of occurrences of w_j in the BoW representation of e_i, and the weight $f(x_{ij})$ is aimed at reducing the impact of rare words. The term G_j captures how well the entity embedding is modelling the word w_j. Similar to Derrac and Schockaert (2015), we found that words which are modelled well, i.e. for which the loss term G_j is low, tend to correspond to semantically meaningful properties. The main idea of our method is to learn multiple GloVe embeddings (i.e. experts), where each embedding will be specialized towards a subset of all words. The key challenge is to train these embeddings such that the properties captured by a given embedding form a semantically meaningful facet or domain. For example, when learning a representation of movies, we would expect to see one GloVe expert that focuses on genre (e.g. capturing words such as *horror*, *zombie* or *funny*).

What makes this problem particularly challenging is that properties from different facets are often correlated (e.g. particular actors may be strongly associated with a particular movie genre). This is in accordance with the theory of conceptual spaces, but it means that a strong inductive bias is needed to learn these representations. Following Alshaikh et al. (2019), we rely on pre-trained word vectors to provide this bias. In particular, we rely on the assumption that whenever a word is related to a given facet, words with similar embeddings tend to be related as well. This assumption is less strong than the assumption from (Alshaikh et al., 2019), where each facet was assumed to correspond to a single cluster.

3.1 Model Formulation

If we ignore the weight $f(x_{ij})$, the relationship between least squares regression and the Gaussian distribution makes it easy to see that the GloVe model maximizes the likelihood of the data X (i.e. the matrix of co-occurrence counts x_{ij}) in accordance with the following probabilistic model:

$$\mathcal{L}(X) = \prod_{i,j} \mathcal{G}(x_{ij} \,|\, \mathbf{e_i} \cdot \tilde{\mathbf{w}}_\mathbf{j} + b_i + \tilde{b}_j, \sigma^2)$$

where $\mathcal{G}$ is the Gaussian distribution and the variance σ^2 is an arbitrary strictly positive constant. In our MoE model, each expert makes a different prediction for the mean $\mathbf{e_i} \cdot \tilde{\mathbf{w}}_\mathbf{j} + b_i + \tilde{b}_j$. Let us write $\mathbf{e_i}^k$ for the embedding of entity e_i by the k$^{\text{th}}$ expert. Similarly, $\tilde{\mathbf{w}}_\mathbf{j}^\mathbf{k}$ corresponds to the embedding of word w_j, according to this expert, while b_i^k and $\tilde{b}_j^k$ are the associated bias terms. We write K for the total number of experts. Furthermore, let us write $g(k,j)$ for the probability that word w_j should be assigned to the k$^{\text{th}}$ expert. The aim of our model is then to maximize the following likelihood:

$$\mathcal{L}(X) = \prod_{i,j} \sum_{k} g(k,j) \mathcal{G}(x_{ij} \,|\, \mathbf{e}_\mathbf{i}^\mathbf{k} \cdot \tilde{\mathbf{w}}_\mathbf{j}^\mathbf{k} + b_i^k + \tilde{b}_j^k, \sigma^2)$$

The probability $g(k,j)$ will be parameterized by a neural network, called the gating network. In particular, let $(y_1^j, ..., y_K^j) = \phi(\mathbf{x_j})$ be the output of a multi-layer perceptron, where the input $\mathbf{x_j}$ is the pre-trained word vector for w_j. The probabilities $g(k,j)$ are then obtained using softmax:

$$g(k,j) = \frac{\exp(y_k^j)}{\sum_{i=1}^{K} \exp(y_i^j)} \tag{2}$$

Note that the decision on which expert should be used for the prediction of x_{ij} only depends on the word w_j in our model. The aim of the gating network is thus to find a meaningful grouping of the words from the BoW representations. Another possibility would be to design the gating network such that the entity e_i is taken into account as well. In principle, this would be useful to determine for each of the learned facets, which entities can have a meaningful representation in that facet. However, in preliminary experiments we were not able to achieve better results with such an approach.

Dataset	Entities	Attribute	Classes
Locations	209121	CORINE land cover level 1 (CL1)	5
		CORINE land cover level2 (CL2)	15
		CORINE land cover level3 (CL3)	44
Movies	13978	Plot keywords (KeyW)	100
		Genre	23
		Age Ratings (AR)	6
Place types	1383	Foursquare categories (Fours.)	9
		Geonames categories (Geo.)	7
		OpenCYC categories (OpenC.)	20

Dataset	Entities	Attribute	Classes
Buildings	3721	Country	2
		Administrative Location(AL)	2
Wikipedia	100000	Semantic Type (SM)	13
		Movies: Language (MoL)	3
		Movies: Color (Mocl)	2
		Movies: Country (MoC)	8
		Music: Country (MuC)	9
		Music: Genre (MuG)	13
		Business: Country (BC)	3
		Business: Legal-form (BLF)	7
		Human: Gender (HG)	2
		Human: Country of citizenship (HC)	2

Table 1: Overview of considered datasets.

3.2 Parameter Estimation

Our aim is now to train the parameters of the gating network and those of the different GloVe experts. We rely on Expectation Maximization (EM) for this purpose.

E-step: For each context word w_j, we estimate a probability distribution over experts, which is based on how well these experts are currently modelling this word. In particular, let us write $\epsilon_{(k,j)}$ for the error term associated with w_j and the k$^{\text{th}}$ expert, i.e.:

$$\epsilon_{(k,j)} = \sum_{i:x_{ij}>0} \left(\mathbf{e_i^k} \cdot \tilde{\mathbf{w}}_\mathbf{j}^\mathbf{k} + b_i^k + \tilde{b}_j^k - \log x_{ij}\right)^2$$

Note that in contrast to the standard GloVe formulation, we do not use the weight $f(x_{ij})$, as we found this weighting strategy not to be helpful in our setting, and omitting it simplifies the formulation of the model. The probability $S_{(k,j)}$ that w_j should be assigned to the k$^{\text{th}}$ expert is then estimated as follows:

$$S_{(k,j)} = \frac{\exp(-\epsilon_{(k,j)})}{\sum_{i\in K}\exp(-\epsilon_{(i,j)})}$$

These probabilities $S_{(k,j)}$ will be used as the supervision signal for training the gating network.
M-step: We train the gating network by minimizing the cross-entropy between the probabilities $S_{(k,j)}$ obtained from the E-step and the probabilities $g(k,j)$ predicted by the gating network:

$$E_{gate} = -\sum_{j=1}^{m}\sum_{k=1}^{K} S_{(k,j)} \ln g(k,j)$$

with $g(k,j)$ defined as in (2). For each expert, the corresponding parameters are learned by using the following weighted version of the standard GloVe loss (without the weights $f(x_{ij})$:

$$G_{(k)} = \sum_{j=1}^{m}\sum_{i:x_{ij}>0} g(k,j)\left(\mathbf{e_i^k} \cdot \tilde{\mathbf{w}}_\mathbf{j}^\mathbf{k} + b_i^k + \tilde{b}_j^k - \log x_{ij}\right)^2$$

In the first iteration of the EM method, the parameters are initialised by training a standard GloVe embedding. In subsequent iterations, we use the parameters from the previous iteration for initialization.

4 Experiments

We experimentally analyze the performance of the proposed mixture-of-experts (MoE) model. Our main focus is on showing that learning facet-specific embeddings is useful compared to learning standard embeddings. We also compare our method with the approach from Alshaikh et al. (2019).

Datasets. We use the *Movies* and *Place types* datasets from Derrac and Schockaert (2015) and the *Buildings* dataset from Alshaikh et al. (2019). These datasets respectively contain BoW descriptions of movies (obtained from reviews), place types (obtained from Flickr tags) and buildings (obtained from Wikipedia articles). Each of these datasets is associated with a number of classification problems, which are listed in Table 1. We refer to the original papers for more details. The aforementioned datasets are all relatively small, since they were used in combination with multi-dimensional scaling in past work. We also evaluate our method on two larger datasets. First, we use a dataset from Jeawak et al. (2019), referred to as *Locations*, in which the entities correspond to geographic locations across the UK and the BoW representations are composed of the tags that were assigned to Flickr photos near these locations. Noting that Flickr tags often correspond to concatenations of different words, we have tokenized these tags using Wordninja (Anderson, 2019), which splits terms based on English Wikipedia unigram frequencies. Subsequently we discarded stop words, using NLTK (Bird and Loper, 2004), as well as words for which we do not have a pre-trained word vector. The classification task associated with this dataset is to predict the CORINE[1] land cover classes at level 1 (5 classes), level 2 (15 classes) and level 3 (44 classes). Second, we compiled a new dataset from the English Wikipedia. In particular, we selected the 100 000 Wikipedia concepts with the longest articles, which approximately corresponds to those concepts whose Wikipedia article contains more than 200 words, after removing stop words and words that appear less than 10 times in the collection. As classification tasks for the Wikipedia dataset, we first consider the problem of predicting the Wikidata semantic type of the Wikipedia entities. In particular, we identified 13 semantic types that occur sufficiently frequently, each having at least 2000 instances in our collection. In addition to these semantic types, we extracted nine attributes from Wikidata, for which a value was specified for a sufficient number of instances: three attributes for *movie* entities and two attributes for each of *music*, *business* and *human*. The considered classification problems are listed in Table 1 [2].

Methodology. For the classification experiments with the *Buildings* and *Place type* datasets, we used 2/3 of the labelled data for training and 1/3 for testing, using the same splits as Alshaikh et al. (2019). For tuning, we use 3-fold cross-validation over the training data. For the other datasets, where we have more labeled data, we split the examples into 60% for training, 20% for tuning and 20% for testing. In the case of the *Movies* dataset, we again used the same split as Alshaikh et al. (2019). To learn the embeddings with our proposed model, we train k experts, choosing k from $\{4, 5, 10\}$ based on the tuning data. In all cases, we fix the total number of dimensions of all embeddings to 100 (e.g. if $k = 4$ then each expert learns a 25-dimensional embedding). As input to the gating network, we use a 50-dimensional GloVe word embedding, which was pre-trained on the English Wikipedia. We run the EM algorithm for five iterations, which we found sufficient for the experts to converge. In each iteration, we train the gating network for 20 epochs and the experts for 10 epochs, using AdagradOptimizer with a learning rate of 0.05 and mini-batch size of 1000. Since the BoW representations from the *Wikipedia* dataset were highly sparse, we used a version of GloVe with negative samples, following (Jeawak et al., 2019).

Baselines. Our main baseline is the standard GloVe model, as in (1), which was found by Jameel and Schockaert (2019) to produce highly competitive entity embeddings, compared to a wide range of other methods. We also experimented with methods based on variational autoencoders, including the Neural Variational Document Model (Miao et al., 2016), but we were not able to obtain competitive results in this way. We fix the number of dimensions in all entity embeddings to 100. We also compare our MoE method against the approaches from Alshaikh et al. (2019), which are referred to as *IncAggGloVe* and *IncHDBGloVe*. These methods differ only in the clustering algorithms that are used for identifying facets, which are Agglomerative Hierarchical Clustering and HDBSCAN, respectively. Note that in contrast to (Alshaikh et al., 2019), where MDS was used, we apply these methods to a 100-dimensional GloVe embedding, to allow for a more direct comparison. However, we found that these methods were not able to scale to the new *Locations* and *Wikipedia* datasets, even when using GloVe for the base embedding, hence we can only consider them for *Movies*, *Place types* and *Buildings*.

Evaluation tasks. We evaluate the quality of the learned embeddings based on the performance of a number of different classifiers which use these embeddings as input. In particular, we followed the

[1] `http://www.eea.europa.eu/data-and-maps/data/corine-land-cover-2006-raster-2`

[2] The source code is available online at
`https://github.com/rana-alshaikh/MoEGloVe`

		Place types			Movies			Buildings	
		Fours.	Geo.	OpenC.	KeyW.	Genre	AR	Country	AL.
DT1	GloVe	0.34	0.23	0.26	0.22	0.32	0.39	0.46	0.30
	IncHDBGloVe	0.35	0.26	0.29	0.24	0.30	0.39	0.48	0.44
	IncAggGloVe	0.35	0.26	0.29	0.23	0.37	0.40	**0.52**	**0.49**
	MoEGloVe	**0.45**	**0.27**	**0.30**	**0.26**	**0.40**	**0.44**	0.48	0.45
DT3	GloVe	0.47	0.29	0.30	0.24	0.37	0.40	0.50	0.41
	IncHDBGloVe	0.50	0.24	0.31	0.24	0.35	0.40	0.52	0.47
	IncAggGloVe	0.44	0.26	0.29	0.23	0.37	0.39	**0.53**	**0.49**
	MoEGloVe	**0.59**	**0.31**	**0.34**	**0.26**	**0.40**	**0.45**	0.52	0.48
SVM	GloVe	0.75	0.39	0.37	0.17	**0.47**	0.33	0.64	0.45
	IncHDBGloVe	0.58	0.27	0.33	0.20	0.44	0.37	0.55	0.51
	IncAggGloVe	0.62	0.26	0.31	0.22	0.46	0.40	0.60	0.50
	MoEGloVe	**0.76**	**0.45**	**0.40**	**0.26**	**0.47**	**0.47**	**0.68**	**0.55**
KNN	GloVe	0.70	0.40	0.38	0.13	0.23	0.30	0.56	0.54
	IncHDBGloVe	0.58	0.38	0.38	0.16	0.33	0.30	0.57	0.52
	IncAggGloVe	0.68	0.32	0.40	0.17	0.32	0.35	0.57	0.50
	MoEGloVe	**0.73**	**0.42**	**0.42**	**0.27**	**0.47**	**0.48**	**0.64**	**0.57**
Gaussian	GloVe	0.71	0.48	0.43	0.19	0.50	0.42	0.62	**0.56**
	IncHDBGloVe	0.62	0.38	**0.50**	0.22	0.44	0.42	0.62	**0.56**
	IncAggGloVe	0.64	0.43	0.48	0.22	0.48	0.41	0.62	**0.56**
	MoEGloVe	**0.74**	**0.50**	0.49	**0.24**	**0.54**	**0.47**	**0.66**	**0.56**

Table 2: Classification performance in terms of F1 score for *Place types*, *Movies* and *Buildings*.

approach from Alshaikh et al. (2019), which uses four types of classification methods. The first method is to train a (linear) SVM classifier on each of the different facet-specific spaces. The predictions of these classifiers are then used as input to a logistic regression meta-classifier. For the GloVe baseline, we simply train an SVM classifier on the full space. Note that this approach is motivated by the theory of conceptual spaces, which suggests that entities have to be compared using Euclidan distance within domain-specific spaces, with overall similarity then determined as a weighted average of the domain-specific similarities. The second method is based on the same view, but instead of using SVMs we use K nearest neighbors (KNN). The value of K was chosen from $\{1, 3, 5\}$ based on the tuning data. A third method, which is also loosely inspired by conceptual spaces, it to estimate a Gaussian distribution (with a diagonal covariance matrix), in each of the facet-specific spaces. To classify a test example, we then add up the log-probabilities obtained from the facet-specific Gaussians. The example is predicted as positive if the result is above a given threshold, which is estimated using maximum likelihood. The advantage of this method is that we do not need to train a separate meta-classifier. Intuitively, if a given facet is not relevant for the category which we are trying to predict, we can expect the corresponding Gaussian to have a high variance, which means that it will have a low impact on the final result. The fourth classification method is based on low-depth decision trees. The aim is to evaluate to what extent important semantic features can be modelled as vectors. In particular, we first select the N words which are best modelled in the vector space (for each expert), i.e. we choose the words j for which the error term $\epsilon_{(k,j)}$ is minimal. To train the decision trees, we then represent each entity $\mathbf{e}$ by the feature vector $(\mathbf{e} \cdot \tilde{\mathbf{w}}^{\mathbf{k}}_{\mathbf{1}}, ..., \mathbf{e} \cdot \tilde{\mathbf{w}}^{\mathbf{k}}_{\mathbf{N}})$, where we write w_i for the i^{th} word that was selected. For the GloVe baseline, we set $N = 2000$. For the other methods, we select $N = 200$ words from each of the facet-specific spaces. We report the results for decision trees of depth 1 (i.e. trees consisting of a single node) and depth 3. A strong performance on this task suggests that the spaces can be described in terms of interpretable linear features, similar to how conceptual spaces are described in terms of quality dimensions.

Results. The results are summarized in Table 2 for the three smaller datasets and in Table 3 for the two larger datasets. As can be seen from the tables, our model outperforms each of the baselines. Moreover, the improvement over GloVe is substantial in many cases, which clearly shows the usefulness of learning multiple facet-specific vector spaces, rather than a single higher-dimensional space. Our model also outperforms IncAgg and IncHDB, in addition to being much more scalable. In fact, surprisingly, the

		Wikipedia										Locations		
		SM.	MoL.	Mocl.	MoC.	MuC	MuG	BLF	BC	HG	HC	CL1	CL2	CL3
DT1	GloVe	0.19	0.37	0.33	**0.18**	0.15	**0.09**	**0.16**	0.30	0.32	0.35	0.24	0.10	0.05
	MoEGloVe	**0.23**	**0.42**	**0.42**	**0.18**	**0.18**	0.08	0.15	**0.34**	**0.43**	**0.49**	**0.31**	**0.13**	**0.06**
DT3	GloVe	0.23	0.38	0.31	0.19	0.16	**0.10**	**0.17**	0.31	0.33	0.36	0.30	0.11	0.05
	MoEGloVe	**0.28**	**0.46**	**0.43**	**0.21**	**0.19**	**0.10**	**0.17**	**0.33**	**0.36**	**0.43**	**0.31**	**0.13**	**0.07**
SVM	GloVe	0.49	0.53	0.48	0.20	0.17	0.08	0.11	0.32	0.44	**0.66**	0.15	0.07	0.01
	MoEGloVe	**0.61**	**0.62**	**0.51**	**0.25**	**0.27**	**0.12**	**0.18**	**0.47**	**0.47**	0.64	**0.30**	**0.10**	**0.06**
KNN	GloVe	0.49	0.33	**0.56**	0.17	0.14	0.07	0.12	0.29	0.50	0.52	0.29	0.21	0.17
	MoEGloVe	**0.60**	**0.50**	0.55	**0.25**	**0.25**	**0.11**	**0.16**	**0.39**	**0.52**	**0.58**	**0.39**	**0.24**	**0.21**
Gauss	GloVe	0.51	0.52	0.55	0.25	0.24	0.11	0.16	**0.19**	0.49	0.13	0.30	0.11	0.10
	MoEGloVe	**0.57**	**0.59**	**0.57**	**0.27**	**0.26**	**0.12**	**0.20**	0.18	**0.51**	**0.16**	**0.33**	**0.17**	**0.14**

Table 3: Classification performance in terms of F1 score For *Wikipedia* and *Locations*.

	Movies	
Movie	**5NN in full space**	**5NN in the Genre facet**
Troy 2004 [Drama, Adventure]	**Alexander 2004** [Action, Drama, Adventure, Biography, War, Romance, History], **Hum 1991** [Action, Drama, Crime, Family], Pig 2010 [Horror], **Kingdom of Heaven 2005** [Action, Drama, Adventure, War, History], **Mac 1992** [Drama]	**Alexander 2004** [Action, Drama, Adventure, Biography, War, Romance, History], **King Arthur 2004** [Action, Drama, Adventure, War, History], **Kingdom of Heaven 2005** [Action, Drama, Adventure, War, History], **Lawrence of Arabia 1962** [Drama, Adventure, Biography, War, History], **Master and Commander, The Far Side of the World 2003** [Action, Drama, Adventure, War]
Iron Man 2008 [Action, Adventure, Sci-Fi]	**Fantastic Four 2005** [Action, Adventure, Fantasy, Sci-Fi, Short], **Hulk 2003** [Action, Sci-Fi], U 2006 [Animation, Family, Music, Musical], **Seven 1979** [Action, Drama],Ali 2001 [Drama, Biography, Sport]	**Hulk 2003** [Action, Sci-Fi], **Aliens 1986** [Action, Adventure, Thriller, Sci-Fi], **Terminator 3-Rise of the Machines 2003** [Action, Thriller, Sci-Fi], **X2 2003** [Action, Adventure, Thriller, Sci-Fi], **Star Trek Nemesis 2002** [Action, Adventure, Thriller, Sci-Fi]
X-Men 2000 [Action, Adventure, Sci-Fi]	**X2 2003** [Action, Adventure, Thriller, Sci-Fi], **Fantastic Four 2005** [Action, Adventure, Fantasy, Sci-Fi, Short], **Spider-Man 2 2004** [Action, Adventure, Fantasy], Nu 2003 [Drama, Short], **ATL 1999** [Comedy, Drama, Crime, Music, Romance]	**Superman II 1980** [Action, Sci-Fi], **Thunder 1983** [Action, Drama, Crime], **X2 2003** [Action, Adventure, Thriller, Sci-Fi], **Iron Man 2008** [Action, Adventure, Sci-Fi], **Terminator 3, Rise of the Machines 2003** [Action, Thriller, Sci-Fi]
The Sound of Music 1965 [Drama, Biography, Family, Music, Musical, Romance]	**Mamma Mia! 2008** [Comedy, Music, Musical, Romance], Show 2003 [Action, Comedy, Crime, Thriller], **Across the Universe 2007** [Drama, Music, Musical, Romance], Pig 2010 [Horror], **Its a Wonderful Life 1946** [Drama, Family, Fantasy]	**Willy Wonka and the Chocolate Factory 1971** [Family, Fantasy, Music, Musical], **Casablanca 1942** [Drama, War, Romance], **Its a Wonderful Life 1946** [Drama, Family, Fantasy], **Singin in the Rain 1952** [Comedy, Music, Musical, Romance], **A Christmas Story 1983** [Comedy, Family]
Mystic River 2003 [Drama, Crime, Thriller, Mystery]	Now 1965 [Music, Short, Documentary], **Training Day 2001** [Action, Drama, Crime, Thriller], **Mac 1992** [Drama], **21 2008** [Drama, Crime, Thriller], **Blue Velvet 1986** [Drama, Crime, Thriller, Mystery]	**21 2008** [Drama, Crime, Thriller], **Atonement 2007** [Drama, War, Mystery, Romance], **Jarhead 2005** [Drama, Biography, War], **The Door 2012** [Horror, Drama, Family, Fantasy, Thriller, Mystery, Sci-Fi, Short], **Red Dragon 2002** [Crime, Thriller]

Table 4: Examples of the Nearest Neighbours from the *Movies* dataset.

IncAgg and IncHDB perform worse than the GloVe baseline in several cases. In contrast, as was reported by Alshaikh et al. (2019), when MDS is used as the base embedding, these methods consistently improve on this base embedding, although they still do not reach the performance of our MoEGLoVe model. A detailed comparison with MDS based representations is provided in the appendix.

While we are not primarily concerned with the overall performance of the classifiers, it is interesting to note that the performance of the SVM, KNN and Gaussian classifiers are broadly comparable. The decision trees perform worse overall, as could be expected. However, the relative performance of the decision trees, compared to the other classifiers, can reveal which categories can be modelled in terms of the most dominant linear features, i.e. the vectors $\mathbf{w}_\mathbf{i}^\mathbf{k}$ with the lowest associated error term $\varepsilon_{(k,j)}$. Such features can intuitively play the role of quality dimensions in applications (Derrac and Schockaert, 2015). The results suggest that the land cover categories in the *Locations* domain correspond to such dominant linear features. In contrast, for the Foursquare categories, the performance of the decision trees is much worse than that of the other classifiers, showing that methods that rely on learned quality dimensions would not model these categories well.

Qualitative Analysis. To illustrate the usefulness of facet-specific vector spaces, Table 4 shows the nearest neighbors of some movies (i) in the full space and (ii) in one of the facet-specific spaces, which is intuitively specialized towards genre. While several of the nearest neighbors in the full space have a similar genre, we can also see many other neighbours (shown in red). In contrast, the neighbors in the

Movies		
Expert 0: animation, soundtrack, studio, remastered, recording, feature, graphics, audio, productions, animated, series, musical, artwork, compilation, version, featuring, interactive, playstation, artists, premiere, theatrical, cinematography, produced, animations, entertainment, unreleased	**Expert 1:** drama, comedy, hollywood, actor, actress, age, teen, children, dramas, mother, remake, horror, death, sitcom, opera, tale, princess, bollywood, shakespeare, wife, broadway, television, fiction, thriller, stories, comedies, romance, family, sex, live, series, documentary, animated, romantic , adventure, mystery, father, fantasy, crime, sequel, reality	**Expert 2:** manuscript, century, medieval, poet, testament, writings, biography, author, memoirs, nineteenth, treatise, philosopher, literature, bestselling, texts, historical, poem, preface, history, narratives, romanticism, allegorical, book, centuries, autobiography, historians, thinkers
Wikipedia		
Expert 0: companies, applications, technologies, firms, equipment, tools, manufacturers,software, motor, computers, management, options, auto, automaker, toyota, product, sale, microsoft, user, buy.	**Expert 1:** music, album, song, radio, film, pop, television, hip-hop, soundtrack, airplay, movies, recordin, tv, billboard, compilation, bbc, chart, musical, band, aired, broadcast, uhf, channel,comic, operas, bands, studio, tunes, indie, broadcasts	**Expert 2:** criminal, crimes, accused, activists, communist, ethnic, opposition, democratic, palestinian, communists, serbs, regime, allies, yugoslavia,israeli, leftist, anti, political
Place types		
Expert 0: italy, thailand, temple, sri, inn, yorkshire, terrace, buddhist, thai, indian, beach, wine, india, cook, malaysia, condo, durham, turkey, village, restaurant, eat, tofu, dining, cornwall	**Expert 1:** ferrari, cessna, jaguar, falcon, musica, mexicana, guitarra, banda, porsche, flight, grupo, airshow, supercars, guitarist, coupe, supercar, peugeot, beetle, jazz,benz, mustang, flamenco, mercedes, amphibian.	**Expert 2:** hair, costumes, cute, kitten, kitty, stockings, dolls, doll, toys, costume, makeup, fashion, nail, puppy, lipstick, teddy, lingerie, smile, sexy, zombie, retro, nails, blouse

Table 5: Examples of the facets from the *Movies*, *Wikipedia* and *Place types* datasets.

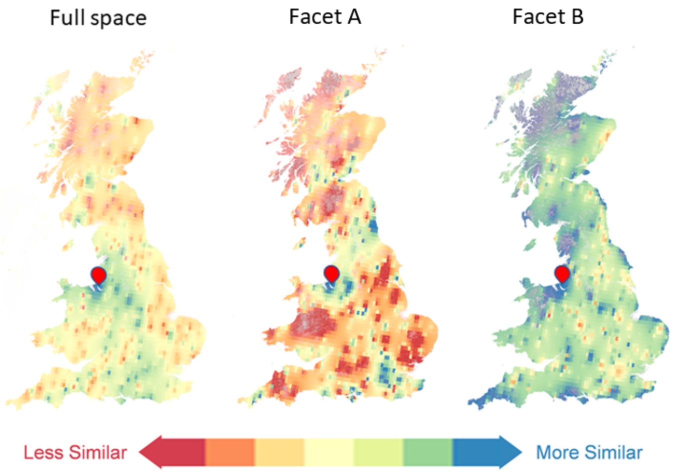

Figure 1: Projection of the full space and two 10-dimensional facets of the *Locations* dataset.

genre-specific space all have a similar genre. In Table 5 we show, for a number of experts, which words are assigned to them by the gating network, i.e. for which words the probability $g(k, j)$ is highest. For the *Movies* domain, for instance, we can see that one expert focused on technical aspects of the movies (e.g. *soundtrack*, *graphics*, *cinematography*), while the second expert focused on genre, and the third expert focused on the particular genre of historical movies. For the *Wikipedia* and *Place types* datasets, which cover a wider range of entities, the discovered facets are mostly thematic. For instance, for the *Wikipedia* dataset, we found facets related to companies, music and politics. Finally, Figure 1 visually shows the different aspects of similarity that are captured by two experts for the *Locations* dataset. Specifically, the figure visualizes how similar different parts of the UK are to the target location, in Liverpool. For one expert (Facet A), the most similar regions correspond to other urban areas (including London, Southampton and Newcastle). On the other hand, the second expert (Facet B) has identified coastal areas across the UK as the most similar regions. While this latter facet may be important in some application contexts, it is clearly not well-captured in the full space (i.e. the standard GloVe embedding).

5 Conclusion

This paper has introduced a method for jointly learning a number of facet-specific low-dimensional entity embeddings. To the best of our knowledge, this is the first approach for learning such representations that is both scalable and unsupervised. We have presented experimental results which show that learning facet-specific spaces can be highly beneficial. While we have focused on bag-of-words input representations in this paper, in future work it would be interesting to see how similar strategies could be applied to document embedding strategies based on BERT (Devlin et al., 2019), or related language models.

Acknowledgements. Steven Schockaert was funded by ERC Starting Grant 637277. Zied Bouraoui was funded by ANR CHAIRE IA BE4musIA.

References

Muhammad Asif Ali, Yifang Sun, Xiaoling Zhou, Wei Wang, and Xiang Zhao. 2019. Antonym-synonym classification based on new sub-space embeddings. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6204–6211.

Carl Allen and Timothy M. Hospedales. 2019. Analogies explained: Towards understanding word embeddings. In *Proceedings of the 36th International Conference on Machine Learning*, pages 223–231.

Rana Alshaikh, Zied Bouraoui, and Steven Schockaert. 2019. Learning conceptual spaces with disentangled facets. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 131–139.

Derek Anderson. 2019. Wordninga: Probabilistically split concatenated words using nlp based on english wikipedia unigram frequencies. `https://github.com/keredson/wordninja`.

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

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

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

Tian Qi Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. 2018. Isolating sources of disentanglement in variational autoencoders. In *Advances in Neural Information Processing Systems*, pages 2610–2620.

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

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186.

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

James W Grau and Deborah K Nelson. 1988. The distinction between integral and separable dimensions: Evidence for the integrality of pitch and loudness. *Journal of Experimental Psychology: General*, 117(4):347–370.

Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2017. An unsupervised neural attention model for aspect extraction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 388–397.

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

Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. 1991. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87.

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

Shoaib Jameel and Steven Schockaert. 2016. Entity embeddings with conceptual subspaces as a basis for plausible reasoning. In *22nd European Conference on Artificial Intelligence*, pages 1353–1361.

Shoaib Jameel and Steven Schockaert. 2019. Word and document embedding with vmf-mixture priors on context word vectors. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*, pages 3319–3328.

Shoaib Jameel, Zied Bouraoui, and Steven Schockaert. 2017. Member: Max-margin based embeddings for entity retrieval. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 783–792.

Shelan S Jeawak, Christopher B Jones, and Steven Schockaert. 2019. Embedding geographic locations for modelling the natural environment using flickr tags and structured data. In *European Conference on Information Retrieval*, pages 51–66. Springer.

Vineet John, Lili Mou, Hareesh Bahuleyan, and Olga Vechtomova. 2019. Disentangled representation learning for non-parallel text style transfer. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*, pages 424–434.

Hyunjik Kim and Andriy Mnih. 2018. Disentangling by factorising. In *Proceedings of the 35th International Conference on Machine Learning*, pages 2654–2663.

Bill Yuchen Lin, Xinyue Chen, Jamin Chen, and Xiang Ren. 2019. Kagnet: Knowledge-aware graph networks for commonsense reasoning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 2829–2839.

Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Rätsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. 2019. Challenging common assumptions in the unsupervised learning of disentangled representations. In *Workshop on Reproducibility in Machine Learning*.

Robert Logan, Nelson F Liu, Matthew E Peters, Matt Gardner, and Sameer Singh. 2019. Barack's wife hillary: Using knowledge graphs for fact-aware language modeling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5962–5971.

Yishu Miao, Lei Yu, and Phil Blunsom. 2016. Neural variational inference for text processing. In *International conference on machine learning*, pages 1727–1736.

Robert M Nosofsky and Thomas J Palmeri. 1996. Learning to classify integral-dimension stimuli. *Psychonomic Bulletin & Review*, 3(2):222–226.

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

Sascha Rothe and Hinrich Schütze. 2016. Word embedding calculus in meaningful ultradense subspaces. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 512–517.

Gerard Salton. 1973. Recent studies in automatic text analysis and document retrieval. *Journal of the ACM*, 20:258–278.

Christophe Van Gysel, Maarten de Rijke, and Evangelos Kanoulas. 2016. Learning latent vector spaces for product search. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 165–174.

Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. 2016. Collaborative knowledge base embedding for recommender systems. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 353–362.

Li Zhang, Shuo Zhang, and Krisztian Balog. 2019. Table2Vec: Neural word and entity embeddings for table population and retrieval. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1029–1032.

Appendices

A Comparison with MDS Based Representations

Table 6 shows a comparison between the methods from this paper and the methods, based on MDS, considered by Alshaikh et al. (2019). For a fair comparison, we relearned the MDS for the movies dataset using only the words that have pre-trained word embedding as they are far less than the number of the total vocabulary.

		Place types			**Movies**			**Buildings**	
		Fours.	Geo.	OpenC.	KeyW.	Genre	AR	Country	AL.
DT1	MDS	0.34	0.26	0.26	**0.26**	0.38	0.43	0.47	0.47
	GloVe	0.34	0.23	0.26	0.22	0.32	0.39	0.46	0.30
	IncAggMDS	**0.45**	**0.30**	**0.30**	0.25	**0.40**	**0.47**	0.50	**0.50**
	IncHDBMDS	0.43	0.26	0.28	0.25	0.38	0.40	0.46	0.46
	IncHDBGloVe	0.35	0.26	0.29	0.24	0.30	0.39	0.48	0.44
	IncAggGloVe	0.35	0.26	0.29	0.23	0.37	0.40	**0.52**	0.49
	MoEGloVe	**0.45**	0.27	**0.30**	**0.26**	**0.40**	0.44	0.48	0.45
DT3	MDS	0.52	0.27	0.32	**0.27**	**0.43**	**0.47**	0.47	0.46
	GloVe	0.47	0.29	0.30	0.24	0.37	0.40	0.50	0.41
	IncAggMDS	0.58	**0.34**	**0.34**	**0.27**	0.41	**0.47**	**0.54**	**0.52**
	IncHDBMDS	0.57	0.26	0.31	0.26	0.39	0.44	0.49	0.50
	IncHDBGloVe	0.50	0.24	0.31	0.24	0.35	0.40	0.52	0.47
	IncAggGloVe	0.44	0.26	0.29	0.23	0.37	0.39	0.53	0.49
	MoEGloVe	**0.59**	0.31	**0.34**	0.26	0.40	0.45	0.52	0.48
SVM	MDS	0.65	0.31	0.35	0.25	0.43	0.45	0.38	0.39
	GloVe	0.75	0.39	0.37	0.17	**0.47**	0.33	0.64	0.45
	IncAggMDS	0.73	0.33	0.37	0.23	**0.47**	0.45	0.52	0.51
	IncHDBMDS	0.65	0.30	0.36	0.23	**0.47**	**0.47**	0.51	0.51
	IncHDBGloVe	0.58	0.27	0.33	0.20	0.44	0.37	0.55	0.51
	IncAggGloVe	0.62	0.26	0.31	0.22	0.46	0.40	0.60	0.50
	MoEGloVe	**0.76**	**0.45**	**0.40**	**0.26**	**0.47**	**0.47**	**0.68**	**0.55**
KNN	MDS	0.65	0.31	0.35	0.20	0.50	0.42	0.47	0.49
	GloVe	0.70	0.40	0.38	0.13	0.23	0.30	0.56	0.54
	IncAggMDS	**0.73**	0.40	0.40	0.25	**0.52**	0.47	0.51	0.50
	IncHDBMDS	0.65	0.33	0.37	0.25	**0.52**	0.25	0.47	0.49
	IncHDBGloVe	0.58	0.38	0.38	0.16	0.33	0.30	0.57	0.52
	IncAggGloVe	0.68	0.32	0.40	0.17	0.32	0.35	0.57	0.50
	MoEGloVe	**0.73**	**0.42**	**0.42**	**0.27**	0.47	**0.48**	**0.64**	**0.57**
Gaussian	MDS	0.81	0.45	0.46	0.26	0.58	0.48	0.53	0.51
	GloVe	0.71	0.48	0.43	0.19	0.50	0.42	0.62	**0.56**
	IncAggMDS	**0.87**	0.48	0.45	0.23	0.55	0.45	0.59	0.55
	IncHDBMDS	0.84	0.43	0.43	0.27	**0.60**	**0.51**	0.54	0.53
	IncHDBGloVe	0.62	0.38	**0.50**	0.22	0.44	0.42	0.62	**0.56**
	IncAggGloVe	0.64	0.43	0.48	0.22	0.48	0.41	0.62	**0.56**
	MoEGloVe	0.74	**0.50**	0.49	**0.24**	0.54	0.47	**0.66**	**0.56**

Table 6: Classification tasks performance (in terms of F1 score) when using the MDS space and GloVe Space.