Structural Regularities in Text-based Entity Vector Spaces

Christophe Van Gysel
University of Amsterdam
Amsterdam, The Netherlands
cvangysel@uva.nl

Maarten de Rijke
University of Amsterdam
Amsterdam, The Netherlands
derijke@uva.nl

Evangelos Kanoulas
University of Amsterdam
Amsterdam, The Netherlands
e.kanoulas@uva.nl

ABSTRACT

Entity retrieval is the task of finding entities such as people or products in response to a query, based solely on the textual documents they are associated with. Recent semantic entity retrieval algorithms represent queries and experts in finite-dimensional vector spaces, where both are constructed from text sequences.

We investigate entity vector spaces and the degree to which they capture structural regularities. Such vector spaces are constructed in an unsupervised manner without explicit information about structural aspects. For concreteness, we address these questions for a specific type of entity: experts in the context of expert finding. We discover how clusterings of experts correspond to committees in organizations, the ability of expert representations to encode the co-author graph, and the degree to which they encode academic rank. We compare latent, continuous representations created using methods based on distributional semantics (LSI), topic models (LDA) and neural networks (word2vec, doc2vec, SERT). Vector spaces created using neural methods, such as doc2vec and SERT, systematically perform better at clustering than LSI, LDA and word2vec. When it comes to encoding entity relations, SERT performs best.

CCS CONCEPTS

•Information systems → Content analysis and feature selection;

ACM Reference format:
Christophe Van Gysel, Maarten de Rijke, and Evangelos Kanoulas. 2016. Structural Regularities in Text-based Entity Vector Spaces. In *Proceedings of ICTIR '17, Amsterdam, Netherlands, October 01-04, 2017*, 8 pages.
DOI: https://doi.org/10.1145/3121050.3121066

1 INTRODUCTION

The construction of latent entity representations is a recurring problem [11, 14, 19, 24, 60] in natural language processing and information retrieval. So far, entity representations are mostly learned from relations between entities [11, 60] for a particular task in a supervised setting [24]. How can we learn latent entity representations if (i) entities only have relations to documents in contrast to other entities (e.g., scholars are represented by the papers they authored), and (ii) there is a lack of labeled data?

As entities are characterized by documents that consist of words, can we use word embeddings to construct a latent entity representation? Distributed representations of words [25], i.e., word embeddings, are learned as part of a neural language model and have been shown to capture semantic [15] and syntactic regularities [39, 43]. In addition, word embeddings have proven to be useful as feature vectors for natural language processing tasks [51], where they have been shown to outperform representations based on frequentist distributional semantics [7]. A downside of word embeddings [8] is that they do not take into account the document a word sequence occurred in or the entity that generated it.

Le and Mikolov [29] address this problem by extending word2vec models to doc2vec by additionally modeling the document a phrase occurred in. That is, besides word embeddings they learn embeddings for documents as well. We can apply doc2vec to the entity representation problem by representing an entity as a pseudo-document consisting of all documents the entity is associated with. Recent advances in entity retrieval incorporate real-world structural relations between represented entities even though the representations are learned from text only. Van Gysel et al. [56] introduce a neural retrieval model (SERT) for an entity retrieval task. In addition to word embeddings, they learn representations for entities.

In this paper, we study the regularities contained within entity representations that are estimated, in an unsupervised manner, from texts and associations alone. Do they correspond to structural real-world relations between the represented entities? E.g., if the entities we represent are people, do these regularities correspond to collaborative and hierarchical structures in their domain (industrial, governmental or academic organizations in the case of experts)? Answers to these questions are valuable because if they allow us to better understand the inner workings of entity retrieval models and give important insights into the entity-oriented tasks they are used for [29]. In addition, future work can build upon these insights to extract structure within entity domains given only a document collection and entity-document relations so to complement or support structured information.

Our working hypothesis is that text-based entity representations encode regularities within their domain. To test this hypothesis we compare latent text-based entity representations learned by neural networks (word2vec, doc2vec, SERT), count-based entity vector representations constructed using Latent Semantic Indexing (LSI) and Latent Dirichlet Allocation (LDA), dimensionality-reduced adjacency representations (Graph PCA) and random representations sampled from a standard multivariate normal distribution. For evaluation purposes we focus on expert finding, a particular case of entity ranking. Expert finding is the task of finding the right person with the appropriate skills or knowledge [5], based on a document collection and associations between people and documents. These associations can be extracted using entity linking methods or from

ICTIR '17, Amsterdam, Netherlands

978-1-4503-4490-6/17/10…$15.00
DOI: https://doi.org/10.1145/3121050.3121066

document meta-data (e.g., authorship). Typical queries are descriptions of expertise areas, such as *distributed computing*, and expert search engines answer the question "Who are experts on *distributed computing*?" asked by people unfamiliar with the field.

Our main finding is that, indeed, semantic entity representations encode domain regularities. Entity representations can be used as feature vectors for clustering and that partitions correspond to structural groups within the entity domain. We also find that similarity between entity representations correlates with relations between entities. In particular, we show how representations of experts in the academic domain encode the co-author graph. Lastly, we show that one of the semantic representation learning methods, SERT, additionally encodes importance amongst entities and, more specifically, the hierarchy of scholars in academic institutions.

2 RELATED WORK

2.1 Representations and regularities

The idea that representations may capture linguistic and semantic regularities or even stereotyped biases that reflect everyday human culture has received considerable attention [13]. The idea of learning a representation of the elements of a discrete set of objects (e.g., words) is not new [25, 46]. However, it has only been since the turn of the last century that Neural Probabilistic Language Models (NPLM), which learn word embeddings as a side-effect of dealing with high-dimensionality, were shown to be more effective than Markovian models [8].

Even more recently, Collobert and Weston explain how the ideas behind NPLMs can be applied to arbitrary Natural Language Processing (NLP) tasks, by learning one set of word representations in a multi-task and semi-supervised setting. Turian et al. [51] compare word representations learned by neural networks, distributional semantics and cluster-based methods as features in Named Entity Recognition (NER) and chunking. They find that both cluster-based methods and distributed word representations learned by NPLMs improve performance, although cluster-based methods yield better representations for infrequent words. Baroni et al. [7] confirm the superiority of context-predicting (word embeddings) over context-counting (distributional semantics) representations.

Later algorithms are specifically designed for learning word embeddings [39, 43], such that, somewhat ironically, NPLMs became a side-product. These embeddings contain linguistic regularities [31, 40], as evidenced in syntactic analogy and semantic similarity tasks. Multiple *word* representations can be combined to form *phrase* representations [38]. Clusterings of word embeddings can be used to discover word classes [38]. And insights gathered from word embedding algorithms can be used to improve distributional semantics [30].

2.2 Entity retrieval

Around 40% of web queries [44] and over 90% of academic search queries [33] concern entities. Entity-oriented queries express an information need that is better answered by returning specific entities as opposed to documents [6]. The entity retrieval task is characterized by a combination of (noisy) textual data and semi-structured knowledge graphs that encode relations between entities [21].

As a particular instance of entity retrieval, expert finding became popular with the TREC Enterprise Track [50]. The task encompasses the retrieval of experts instead of documents. This is useful in enterprise settings, where employers seek to facilitate information exchange and stimulate collaboration [17]. Expert finding diverges from the generic entity retrieval task due to the lack of explicit relations between experts. Balog et al. [2] introduce language models for expert finding. In the maximum-likelihood language modeling paradigm, experts are represented as a normalized bag-of-words vector with additional smoothing. These vectors are high-dimensional and sparse due to the large vocabularies used in expert domains. Therefore, bag-of-words vectors are unsuited for use as representations as lower-dimensional and continuous vector spaces are preferred in machine learning algorithms [59]. Demartini et al. [19] introduce a framework for using document vector spaces in expert finding. Fang et al. [22] explore the viability of learning-to-rank methods in expert retrieval. van Dijk et al. [52] propose methods for detecting potential experts in community question-answering.

Van Gysel et al. [54, 56] propose a neural language modeling approach to expert finding; they also release the Semantic Entity Retrieval Toolkit (SERT) that we use in this paper. Closely related to expert finding is the task of expert profiling, of which the goal is to describe an expert by her areas of expertise [3], and similar expert finding [4]; see [5] for an overview.

2.3 Latent semantic information retrieval

The mismatch between queries and documents is a critical challenge in search [32]. Latent Semantic Models (LSMs) retrieve objects based on conceptual, or semantic, rather than exact word matches. The introduction of Latent Semantic Indexing (LSI) [18], followed by probabilistic LSI (pLSI) [26], led to an increase in the popularity of LSMs. Salakhutdinov and Hinton [47] perform unsupervised learning of latent semantic document bit patterns using a deep auto-encoder. Huang et al. introduced Deep Structured Semantic Models [27, 49] that predict a document's relevance to a query using click data. Neural network models have also been used for learning to rank [12, 20, 34].

3 TEXT-BASED ENTITY VECTOR SPACES

For text-based entity retrieval tasks we are given a document collection D and a set of entities X. Documents $d \in D$ consist of a sequence of words $w_1, \ldots, w_{|d|}$ originating from a vocabulary V, where $|\cdot|$ denotes the document length in number of words. For every document d we have a set $X_d \subset X$ of associated entities (X_d can be empty for some documents) and conversely $D_x \subset D$ consists of all documents associated with entity x. The associations between documents and experts can be obtained in multiple ways. E.g., named-entity recognition can be applied to the documents and mentions can subsequently be linked to entities. Or associations can be extracted from document meta-data (e.g., authorship).

Once determined, the associations between entities X and documents D encode a bipartite graph. If two entities $x_i, x_j \in X$ are associated with the same document, we say that x_i and x_j are co-associated. However, the semantics of a co-association are equivocal as the semantics of an association are ambiguous by itself (e.g.,

author vs. editor). Therefore, instead of relying solely on document associations, we use the textual data of associated documents to construct an entity representation.

Vector space models for document retrieval, such as LSI [18] or LDA [10], can be adapted to entity retrieval. We substantiate this for a specific entity retrieval task: expert finding. As there are many more documents than experts, it is not ideal to estimate a vector space directly on the expert-level using bag-of-word vectors (e.g., by representing every expert as a concatenation of its documents) due to data sparsity. Therefore, it is preferable to first estimate a vector space on the document collection and then use the obtained document representations to construct an entity vector. Demartini et al. [19] take an entity's representation to be the sum of its documents:

$$e_i = \sum_{d_j \in D_{x_i}} g(d_j), \tag{1}$$

where e_i is the k-dimensional vector representation of entity $x_i \in X$ and g is the function mapping a document to its vector space representation (e.g., LSI). The dimensionality k depends on the underlying vector space. For simple bag-of-words representations, k is equal to the number of words in the vocabulary. For latent vector spaces (e.g., LSI), the k-dimensional space encodes latent concepts and the choice of k is left to the user.

Vector space models for document retrieval are often constructed heuristically. E.g., Eq. 1 does not make optimal use of document-entity associations as document representations are added without taking into consideration the significance of words contained within them [35]. And if many diverse documents are associated with an expert, then Eq. 1 is likely to succumb to the noise in these vectors and yield meaningless representations.

To address this problem, Le and Mikolov [29] introduced doc2vec by adapting the word2vec models to incorporate the document a phrase occurs in. They optimize word and document embeddings jointly to predict a word given its context and the document the word occurs in. The key difference between word2vec and doc2vec is that the latter considers an additional meta-token in the context that represents the document. Instead of performing dimensionality reduction on bag-of-words representations, doc2vec learns representations from word phrases. Therefore, we use the doc2vec model to learn expert embeddings by representing every expert $x_j \in X$ as a pseudo-document consisting of the concatenation of their associated documents D_{x_j}.

A different neural language model architecture than doc2vec was proposed by Van Gysel et al. [56], specifically for the expert finding task. For a given word w_i and expert x_j:

$$\text{score}(w_i, x_j) = \exp\left(v_i^\intercal \cdot e_j + b_j\right), \tag{2}$$

where v_i (e_j, resp.) are the latent k-dimensional representations of word w_i (and expert x_j, respectively) and b_j is the bias scalar associated with expert x_j. Eq. 2 can be interpreted as the unnormalized factor product of likelihood $P(w_i \mid x_j)$ and prior $P(x_j)$ in log-space. The score is then transformed to the conditional probability

$$P(X = x_j \mid w_i) = \frac{\text{score}(w_i, x_j)}{\sum_{x_l \in X} \text{score}(w_i, x_l)}.$$

Unlike Eq. 1, the conditional probability distribution $P(X = x_j \mid w_i)$ will be skewed towards relevant experts if the word w_i is significant as described by Luhn [35]. The parameters v_i, e_j and b_j are learned from the corpus using gradient descent. See [56] for details.

Our focus lies on representations of entities e_j and how these correspond to structures within their domains (i.e., organizations for experts). These representations are estimated using a corpus only and can be interpreted as vectors in word embedding space that correspond to entities (i.e., people) instead of words.

4 EXPERIMENTAL SET-UP

4.1 Research questions

We investigate regularities within text-based entity vector spaces, using expert finding as our concrete test case, and ask how these representations correspond to structure in their respective domains. We seek to answer the following research questions:

RQ1 Do clusterings of text-based entity representations reflect the structure of their domains?

Many organizations consist of smaller groups, committees or teams of experts who are appointed with a specific role. When we cluster expert representations, do the clusters correspond to these groups?

RQ2 To what extent do different text-based entity representation methods encode relations between entities?

The associations within expert domains encode a co-association graph structure. To what extent do the different expertise models encode this co-association between experts? In particular, if we rank experts according to their nearest neighbors, how does this ranking correspond to the academic co-author graph?

4.2 Expert finding collections

We use publicly-available expert finding collections provided by the World Wide Web Consortium (W3C) and Tilburg University (TU); see Table 1.

W3C. The W3C collection was released as part of the 2005–2006 editions of the TREC Enterprise Track [16]. It contains a heterogeneous crawl of W3C's website (June 2004) and consists of mailing lists and discussion boards among others. In the 2005 edition, TREC released a list of working groups and their members. Each working group is appointed to study and report on a particular aspect of the World Wide Web to enable the W3C to pursue its mission. We use the associations provided by Van Gysel et al. [56], which they gathered by applying named entity recognition and linking these mentions to a list of organization members, as proposed by Balog et al. [2].

TU. The TU collection consists of a crawl of a university's internal website and contains bi-lingual documents, such as academic publications, course descriptions and personal websites [9]. The document-candidate associations are part of the collection. For every member of the academic staff, their academic title is included as part of the collection.

Table 1: An overview of the two expert finding collections (W3C and TU).

	W3C	TU
Documents in collection	331,037	31,209
Average tokens per document	1,237.23	2,454.93
Number of candidate experts	715	977
Number of document-candidate associations	200,939	36,566
Number of documents (with $\|X_d\| > 0$)	93,826	27,834
Number of associations per document	2.14 ± 3.29	1.13 ± 0.39
Number of associations per candidate	281.03 ± 666.63	37.43 ± 61.00

4.3 Implementations and parameters

We follow a similar experimental set-up as previous work [2, 19, 38, 56]. For LSI, LDA, word2vec and doc2vec we use the Gensim[1] implementation, while for the log-linear model we use the Semantic Entity Retrieval Toolkit[2] (SERT) [55].

The corpora are normalized by lowercasing and removing punctuation and numbers. The vocabulary is pruned by removing stop words and retaining the 60k most frequent words. We sweep exponentially over the vector space dimensionality (k = 32, 64, 128 and 256) of the methods under comparison. This allows us to evaluate the effect of differently-sized vector spaces and their modeling capabilities.

For word2vec, a query/document is represented by its average word vector, which is effective for computing short text similarity [28]. We report both Continuous Bag-of-Words (CBOW) and Skip-gram (SG) variants of word2vec. For LDA, we set $\alpha = \beta = 0.1$ and train the model for 100 iterations or until topic convergence is achieved. Unlike Van Gysel et al. [56], for SERT, we do not initialize with pre-trained word2vec embeddings. Default parameters are used in all other cases.

For LSI, LDA and word2vec, expert representations are created from document representations according to Eq. 1.

In addition to text-based representations, we also include two baselines that do not consider textual data. For the first method (Graph PCA), we construct a weighted, undirected co-association graph where the weight between two entities is given by the number of times they are co-associated. We then apply Principal Component Analysis to create a latent representation for every entity. Secondly, we include a baseline where experts are represented as a random vector sampled from a standard multivariate normal distribution.

5 REGULARITIES IN ENTITY VECTOR SPACES

We investigate regularities within latent text-based entity vector spaces. In particular, we first build latent representations for experts and ground these in the structure of the organizations where these experts are active. First, we cluster latent expert representations using different clustering techniques and compare the resulting clusters to committees in a standards organization of the World Wide Web (RQ1). We continue by investigating to what extent these representations encode entity relations (RQ2). We complement the answers to our research questions with an analysis of the prior (the scalar bias in Eq. 2) associated with every expert in one of the models we consider, SERT, and compare this to their academic rank.

5.1 Answers to research questions

RQ1 Do clusterings of text-based entity representations reflect the structure of their domains?

The World Wide Web Consortium (W3C) consists of various working groups.[3] Each working group is responsible for a particular aspect of the WWW and consists of two or more experts. We use these working groups as ground truth for evaluating the ability of expert representations to encode similarity. The W3C working groups are special committees that are established to produce a particular deliverable [45, p. 492] and are a way to gather experts from around the organization who share areas of expertise and who would otherwise not directly communicate. Working groups are non-hierarchical in nature and represent clusters of experts. Therefore, they can be used to evaluate to what extent entity representations can be used as feature vectors for clustering.

We cluster expert representations using K-means [36]. While K-means imposes strong assumptions on cluster shapes (convexity and isotropism), it is still very popular today due to its linear time complexity, geometric interpretation and absence of hard to choose hyper-parameters (unlike spectral variants or DBSCAN). We cluster expert representations of increasing dimensionality k ($k = 2^i$ for $5 \leq i < 9$) using a linear sweep over the number of clusters K ($10^0 \leq K < 10^2$).

During evaluation we transform working group memberships to a hard clustering of experts by assigning every expert to the smallest working group to which they belong as we wish to find specialized clusters contrary to general clusters that contain many experts. We then use Adjusted Mutual Information, an adjusted-for-chance variant of Normalized Information Distance [58], to compare both clusterings. Adjusting for chance is important as non-adjusted measures (such as BCubed precision/recall[4] as presented

[1] https://radimrehurek.com/gensim
[2] https://github.com/cvangysel/SERT
[3] http://www.w3.org/Consortium/activities
[4] This can be verified empirically by computing BCubed measures for an increasing number of random partitions.

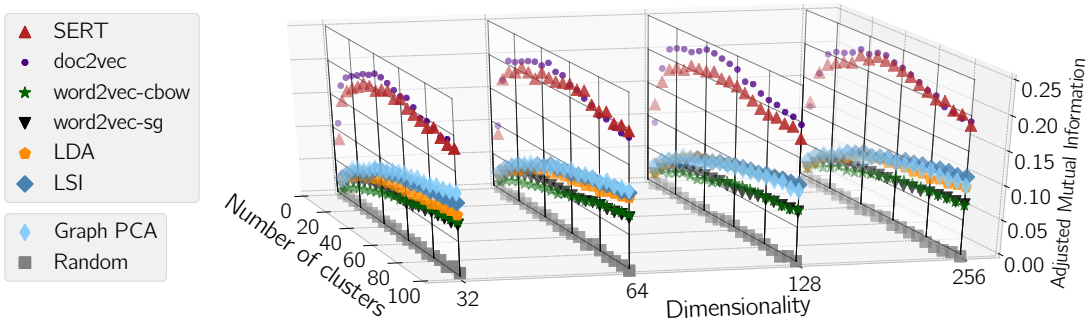

Figure 1: Comparison of clustering capabilities of expert representations (random, Graph PCA, LSI, LDA, word2vec, doc2vec and SERT) using K-means for $10^0 \leq K < 10^2$ (y-axis). The x-axis shows the dimensionality of the representations and the z-axis denotes the Adjusted Mutual Information.

by Amigó et al. [1]) have the tendency to take on a higher value for a larger value of K. Performing the adjustment allows us to compare clusterings for different values of K. We repeat the K-means clustering 10 times with different centroids initializations and report the average.

Figure 1 shows the clustering capabilities of the different representations for different values of K and vector space dimensionality. Ignoring the random baseline, representations built using word2vec perform worst. This is most likely due to the fact that document representations for word2vec are constructed by averaging individual word vectors. Next up, we observe a tie between LSI and LDA. Interestingly enough, the baseline that only considers entity-document associations and does not take into account textual content, Graph PCA, outperforms all representations constructed from document-level vector space models (Eq. 1). Furthermore, doc2vec and SERT perform best, regardless of vector space dimensionality, and consistently outperform the other representations. If we look at the vector space dimensionality, we see that the best clustering is created using 128-dimensional vector spaces. Considering the number of clusters, we see that doc2vec and SERT peak at about 40 to 60 clusters. This corresponds closely to the number of ground-truth clusters. The remaining representations (word2vec, LSI, LDA, Graph PCA) only seem to plateau in terms of clustering performance at $K = 100$, far below the clustering performance of the doc2vec and SERT representation methods.

To answer our first research question, we conclude that expert representations can be used to discover structure within organizations. However, the quality of the clustering varies greatly and use of more advanced methods (i.e., doc2vec or SERT) is recommended.

RQ2 To what extent do different text-based entity representation methods encode relations between entities?

The text-based entity representation problem is characterized by a bipartite graph of entities and documents where an edge denotes an entity-document association. This differs from entity finding settings where explicit entity-entity relations are available and fits into the scenario where representations have to be constructed from unstructured text only. If latent text-based entity representations encode co-associations, then we can use this insight for (1) a better understanding of text-based entity representation models, and (2) the usability of latent text-based entity representations as feature vectors in scenarios where relations between entities are important.

We evaluate the capacity of text-based expert representations to encode co-associations by casting the problem as a ranking task. Contrary to typical expert finding, where we rank experts according to their relevance to a textual query, for the purpose of answering RQ2, we rank experts according to their cosine similarity w.r.t. a query expert [4]. This task shares similarity with content-based recommendation based on unstructured data [42].

In expert finding collections, document-expert associations can indicate many things. For example, in the W3C collection, entity-document associations are mined from expert mentions [2]. However, for the TU collection, we know that a subset of associations corresponds to academic paper authorship. Therefore, we construct ranking ground-truth from paper co-authorship and take the relevance label of an expert to be the number of times the expert was a co-author with the query expert (excluding the query expert themselves). Our intention is to determine to what extent latent entity representations estimated from text can reconstruct the original co-author graph. Given that we estimate the latent entity representations using the complete TU document collection, by design, our evaluation is contained within our training set for the purpose of this analysis.

Table 2 shows NDCG and R-Precision [37, p. 158] for various representation models and dimensionality. SERT performs significantly better than the other representations methods (except for the 256-dimensional representations where significance was not achieved w.r.t. LDA). SERT is closely followed by word2vec (of which both variants score only slightly worse than SERT), LDA and LSI. The count-based distributional methods (LSI, LDA) perform better as the dimensionality of the representations increases. This is contrary to SERT, where retrieval performance is very stable across dimensionalities. Interestingly, doc2vec performs very poorly at reconstructing the co-author graph and is even surpassed

Table 2: Retrieval performance (NDCG and R-Precision) when ranking experts for a query expert by the cosine similarity of expert representations (random, Graph PCA, LSI, LDA, word2vec, doc2vec and SERT) for the TU expert collection (§4.2) for an increasing representation dimensionality. The relevance labels are given by the number of times two experts were co-authors of academic papers. Significance of results is determined using a two-tailed paired Student t-test (* $p < 0.10$, ** $p < 0.05$, *** $p < 0.01$) between the best performing model and second best performing method.

Dimensionality k =	32		64		128		256	
	NDCG	R-Precision	NDCG	R-Precision	NDCG	R-Precision	NDCG	R-Precision
Random	0.18	0.01	0.18	0.01	0.18	0.01	0.18	0.01
Graph PCA	0.38	0.18	0.39	0.20	0.41	0.23	0.39	0.23
LSI	0.39	0.17	0.43	0.21	0.46	0.23	0.47	0.23
LDA	0.44	0.19	0.45	0.20	0.46	0.22	0.52	0.28
word2vec-sg	0.46	0.22	0.49	0.24	0.49	0.24	0.50	0.25
word2vec-cbow	0.46	0.23	0.47	0.24	0.48	0.25	0.48	0.25
doc2vec	0.35	0.14	0.36	0.15	0.36	0.16	0.35	0.15
SERT	**0.53*****	**0.29*****	**0.54*****	**0.31*****	**0.53*****	**0.30*****	**0.53**	**0.31***

by the Graph PCA baseline. This is likely due to the fact that doc2vec is trained on expert profiles and is not explicitly presented with document-expert associations. The difference in performance between doc2vec and SERT for RQ2 reflects a difference in architecture: while SERT is directly optimized to discriminate between entities, doc2vec models entities as context in addition to language. Hence, similarities and dissimilarities between entities are preserved much better by SERT.

We answer our second research question as follows. Latent text-based entity representations do encode information about entity relations. However, there is a large difference in the performance of different methods. SERT seems to encode the entity co-associations better than other methods, by achieving the highest performance independent of the vector space dimensionality.

5.2 Analysis of the expert prior in SERT

One of the semantic models that we consider, SERT, learns a prior $P(X)$ over entities. The remaining representation learning methods do not encode an explicit entity prior. It might be possible to extract a prior from generic entity vector spaces, e.g., by examining the deviation from the mean representation for every entity. However, developing such prior extraction methods is a topic of study by itself and is out of scope for this paper.

In the case of expert finding, this prior probability encodes a belief over experts without observing any evidence (i.e., query terms in SERT). Which structural information does this prior capture? We now investigate the regularities encoded within this prior and link it back to the hierarchy among scholars in the Tilburg University collection. We estimate a SERT model on the whole TU collection and extract the prior probabilities:

$$P(X = x_i) = \frac{\exp(b_i)}{\sum_l \exp(b_l)}, \tag{3}$$

where b is the bias vector of the SERT model in Eq. 2.

For 666 out of 977 experts in the TU collection we have ground truth information regarding their academic rank [9].[5] Figure 2 shows box plots of the prior probabilities, learned automatically by the SERT model from only text and associations, grouped by academic rank. Interestingly, the prior seems to encode the hierarchy amongst scholars at Tilburg University, e.g., Post-docs are ranked higher than PhD students. This is not surprising as it is quite likely that higher-ranked scholars have more associated documents.

The prior over experts in SERT encodes rank within organizations. As mentioned earlier, this is not surprising, as experts (i.e., academics in this experiment) of higher rank tend to occur more frequently in the expert collection. This observation unveils interesting insights about the expert finding task and consequently models targeted at solving it. Unlike unsupervised ad-hoc document retrieval where we assume a uniform prior and normalized document lengths, the prior over experts in the expert finding task is of much greater importance. In addition, we can use this insight to gain a better understanding of the formal language models for expertise retrieval [2]. Balog et al. [2] find that, for the expert finding task, the document-oriented language model performs better than a entity-oriented language model. However, the document-oriented model [2] will rank experts with more associated documents higher than experts with few associated documents. On the contrary, the entity-oriented model of Balog et al. [2], imposes a uniform prior over experts. SERT is an entity-oriented model and performs better than the formal document-oriented language model [56]. This is likely due to the fact that SERT learns an empirical prior over entities instead of making an assumption of uniformity, in addition to its entity-oriented perspective.

In the case of general entity finding, the importance of the number of associated documents might be of lesser importance. Other sources of prior information, such as link analysis [41], recency [23] and user interactions [48], can be a better way of modeling entity importance than the length of entity descriptions.

[5] 126 PhD Students, 49 Postdoctoral Researchers, 210 Assistant Professors, 89 Associate Professors and 190 Full Professors; we filtered out academic ranks that only occur once in the ground-truth, namely Scientific Programmer and Research Coordinator.

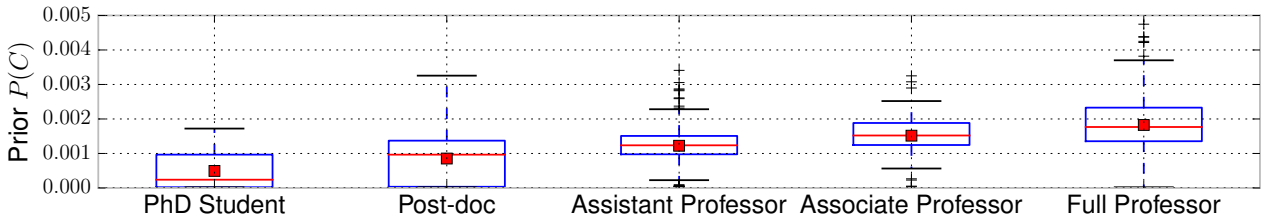

Figure 2: Box plots of prior probabilities learned by SERT, grouped by the experts' academic rank, for the TU collection. We only show the prior learned for a SERT model with $k = 32$, as the distributions of models with a different representation dimensionality are qualitatively similar.

6 CONCLUSIONS

In this work we have investigated the structural regularities contained within latent text-based entity representations. Entity representations were constructed from expert finding collections using methods from distributional semantics (LSI), topic models (LDA) and neural networks (word2vec, doc2vec and SERT). For LSI, LDA and word2vec, document-level representations were transformed to the entity scope according to the framework of Demartini et al. [19]. In the case of doc2vec and SERT, entity representations were learned directly. In addition to representations estimated only from text, we considered non-textual baselines, such as: (1) random representations sampled from a Normal distribution, and (2) the rows of the dimensionality-reduced adjacency matrix of the co-association graph.

We have found that text-based entity representations can be used to discover groups inherent to an organization. We have clustered entity representations using K-means and compared the obtained clusters with a ground-truth partitioning. No information about the organization is presented to the algorithms. Instead, these regularities are extracted by the documents associated with entities and published within the organization. Furthermore, we have evaluated the capacity of text-based expert representations to encode co-associations by casting the problem as a ranking task. We discover that text-based representations retain co-associations up to different extents. In particular, we find that SERT entity representations encode the co-association graph better than the other representation learning methods. We conclude that this is due to the fact that SERT representations are directly optimized to discriminate between entities. Lastly, we have shown that the prior probabilities learned by semantic models encode further structural information. That is, we find that the prior probability over experts (i.e., members of an academic institution), learned as part of a SERT model, encodes academic rank. In addition, we discuss the similarities between SERT and the document-oriented language model [2] and find that the document association prior plays an important role in expert finding.

Our findings have shown insight into how different text-based entity representation methods behave in various applications. In particular, we find that the manner in which entity-document associations are encoded plays an important role. That is, representation learning methods that directly optimize the representation of the entity seem to perform best. When considering different neural representation learning models (doc2vec and SERT), we find that their difference in architecture allows them to encode different regularities: doc2vec models an entity as context in addition to language, whereas SERT learns to discriminate between entities given their language. Thus, doc2vec can more adequately model the topical nature of entities, while SERT more closely captures the similarities and dissimilarities between entities. In the case of expert finding, we find that the amount of textual data associated with an expert is a principal measure of expert importance.

Future work includes the use of text-based entity representations in end-to-end applications. For example, in social networks these methods can be applied to cluster users in addition to network features [53, 57], or to induce graphs based on thread participation or hashtag usage. In addition, text-based entity representations can be used as item feature vectors in recommendation systems. Beyond text-only entity collections, there is also a plenitude of applications where entity relations are available. While there has been some work on learning latent representations from entity relations [11, 60], there has not been much attention given to combining textual evidence and entity relations. Therefore, we identify two additional directions for future work. First, an analysis showing in what capacity entity representations estimated from text alone encode entity-entity relations (beyond the co-associations considered in this work). Secondly, the incorporation of entity-entity similarity in the construction of latent entity representations.

ACKNOWLEDGMENTS

We would like to thank the anonymous reviewers for their valuable comments and suggestions. This research was supported by Ahold Delhaize, Amsterdam Data Science, the Bloomberg Research Grant program, the Criteo Faculty Research Award program, the Dutch national program COMMIT, Elsevier, the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement nr 312827 (VOX-Pol), the Google Faculty Research Award scheme, the Microsoft Research Ph.D. program, the Netherlands Institute for Sound and Vision, the Netherlands Organisation for Scientific Research (NWO) under project nrs 612.001.116, HOR-11-10, CI-14-25, 652.002.001, 612.001.551, 652.001.003, and Yandex. All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.

REFERENCES

[1] Enrique Amigó, Julio Gonzalo, Javier Artiles, and Felisa Verdejo. 2009. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Information Retrieval* 12, 4 (2009), 461–486.

[2] Krisztian Balog, Leif Azzopardi, and Maarten de Rijke. 2006. Formal models for expert finding in enterprise corpora. In *SIGIR*. 43–50.

[3] Krisztian Balog and Maarten de Rijke. 2007. Determining Expert Profiles (With an Application to Expert Finding). In *IJCAI*. 2657–2662.

[4] Krisztian Balog and Maarten de Rijke. 2007. Finding similar experts. In *SIGIR*. ACM, 821–822.

[5] Krisztian Balog, Yi Fang, Maarten de Rijke, Pavel Serdyukov, and Luo Si. 2012. Expertise Retrieval. *Found. & Tr. in Information Retrieval* 6, 2-3 (2012), 127–256.

[6] Krisztian Balog, Pavel Serdyukov, and Arjen P. de Vries. 2010. Overview of the TREC 2010 entity track. In *TREC*. NIST.

[7] Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 2014. Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors.. In *ACL*. 238–247.

[8] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A Neural Probabilistic Language Model. *JMLR* 3 (2003), 1137–1155.

[9] Richard Berendsen, Maarten de Rijke, Krisztian Balog, Toine Bogers, and Antal van den Bosch. 2013. On the Assessment of Expertise Profiles. *JASIST* 64, 10 (2013), 2024–2044.

[10] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet allocation. *JMLR* 3 (2003), 993–1022.

[11] Antoine Bordes, Jason Weston, Ronan Collobert, and Yoshua Bengio. 2011. Learning structured embeddings of knowledge bases. In *AAAI*.

[12] Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. 2005. Learning to rank using gradient descent. In *ICML*. 89–96.

[13] Aylin Caliskan, Joanna J. Bryson, and Arvind Narayanan. 2017. Semantics derived automatically from language corpora contain human-like biases. *Science* 356, 6334 (2017), 183–186.

[14] Kevin Clark and Christopher D Manning. 2016. Improving Coreference Resolution by Learning Entity-Level Distributed Representations. arXiv 1606.01323. (2016).

[15] Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *ICML*. 160–167.

[16] Nick Craswell, Arjen P. de Vries, and Ian Soboroff. 2005. Overview of the TREC 2005 Enterprise Track. In *TREC*.

[17] Thomas H. Davenport and Laurence Prusak. 1998. *Working knowledge: How organizations manage what they know*. Harvard Business Press.

[18] Scott C. Deerwester, Susan T. Dumais, and Richard A. Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science* 41, 6 (1990), 391–407.

[19] Gianluca Demartini, Julien Gaugaz, and Wolfgang Nejdl. 2009. A vector space model for ranking entities and its application to expert search. In *ECIR*. Springer, 189–201.

[20] Li Deng, Xiaodong He, and Jianfeng Gao. 2013. Deep stacking networks for information retrieval. In *ICASSP*. 3153–3157.

[21] Laura Dietz, Alexander Kotov, and Edgar Meij. 2016. Utilizing Knowledge Bases in Text-centric Information Retrieval. In *ICTIR*. ACM, 5–5.

[22] Yi Fang, Luo Si, and Aditya P. Mathur. 2010. Discriminative models of integrating document evidence and document-candidate associations for expert search. In *SIGIR*. ACM, 683–690.

[23] David Graus, Manos Tsagkias, Wouter Weerkamp, Edgar Meij, and Maarten de Rijke. 2016. Dynamic collective entity representations for entity ranking. In *WSDM*. ACM, 595–604.

[24] Zhengyan He, Shujie Liu, Mu Li, Ming Zhou, Longkai Zhang, and Houfeng Wang. 2013. Learning Entity Representation for Entity Disambiguation.. In *ACL*. 30–34.

[25] Geoffrey E. Hinton. 1986. Learning distributed representations of concepts. In *8th Annual Conference of the Cognitive Science Society*, Vol. 1. Amherst, MA, 12.

[26] Thomas Hofmann. 1999. Probabilistic latent semantic indexing. In *SIGIR*. ACM, 50–57.

[27] Po-sen Huang, N Mathews Ave Urbana, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning Deep Structured Semantic Models for Web Search using Clickthrough Data. In *CIKM*. 2333–2338.

[28] Tom Kenter and Maarten de Rijke. 2015. Short text similarity with word embeddings. In *CIKM*. ACM, 1411–1420.

[29] Quoc V Le and Tomas Mikolov. 2014. Distributed Representations of Sentences and Documents. In *ICML*. 1188–1196.

[30] Omer Levy, Yoav Goldberg, and Ido Dagan. 2015. Improving distributional similarity with lessons learned from word embeddings. *TACL* 3 (2015), 211–225.

[31] Omer Levy, Yoav Goldberg, and Israel Ramat-Gan. 2014. Linguistic Regularities in Sparse and Explicit Word Representations. In *CoNLL*. 171–180.

[32] Hang Li and Jun Xu. 2014. Semantic Matching in Search. *Found. & Tr. in Information Retrieval* 7, 5 (June 2014), 343–469.

[33] Xinyi Li, Bob Schijvenaars, and Maarten de Rijke. 2017. Investigating queries and search failures in academic search. *Information Processing & Management* 53, 3 (May 2017), 666–683.

[34] Tie-Yan Liu. 2011. *Learning to Rank for Information Retrieval*. Springer.

[35] Hans Peter Luhn. 1958. The automatic creation of literature abstracts. *IBM Journal of research and development* 2, 2 (1958), 159–165.

[36] James B. MacQueen. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*. 281–297.

[37] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.

[38] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *NIPS*. 3111–3119.

[39] Tomas Mikolov, Greg Corrado, Kai Chen, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. arXiv 1301.3781. (2013).

[40] Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013. Linguistic Regularities in Continuous Space Word Representations. In *HLT-NAACL*. 746–751.

[41] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. *The PageRank citation ranking: bringing order to the web*. Technical Report. Stanford InfoLab.

[42] Michael J Pazzani and Daniel Billsus. 2007. Content-based recommendation systems. In *The adaptive web*. Springer, 325–341.

[43] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. In *EMNLP*. 1532–1543.

[44] Jeffrey Pound, Peter Mika, and Hugo Zaragoza. 2010. Ad-hoc Object Retrieval in the Web of Data. In *WWW*. ACM, 771–780.

[45] Henry Martyn Robert, Sarah Corbin Robert, and Daniel H Honemann. 2011. *Robert's rules of order newly revised*. Da Capo Press.

[46] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. 1985. *Learning internal representations by error propagation*. Technical Report. DTIC Document.

[47] Ruslan Salakhutdinov and Geoffrey Hinton. 2009. Semantic hashing. *Int. J. Approximate Reasoning* 50, 7 (2009), 969–978.

[48] Anne Schuth. 2016. Search Engines That Learn from Their Users. *SIGIR Forum* 50, 1 (June 2016), 95–96.

[49] Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng, and Grégoire Mesnil. 2014. A Latent Semantic Model with Convolutional-Pooling Structure for Information Retrieval. In *CIKM*. 101–110.

[50] TREC. 2005–2008. Enterprise Track. (2005–2008).

[51] Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: a simple and general method for semi-supervised learning. In *ACL*. 384–394.

[52] David van Dijk, Manos Tsagkias, and Maarten de Rijke. 2015. Early Detection of Topical Expertise in Community Question Answering. In *SIGIR*. ACM, 995–998.

[53] Christophe Van Gysel. 2014. Listening to the Flock - Towards opinion mining through data-parallel, semi-supervised learning on social graphs. (2014).

[54] Christophe Van Gysel, Maarten de Rijke, and Evangelos Kanoulas. 2016. Learning Latent Vector Spaces for Product Search. In *CIKM*. ACM, 165–174.

[55] Christophe Van Gysel, Maarten de Rijke, and Evangelos Kanoulas. 2017. Semantic Entity Retrieval Toolkit. In *Neu-IR SIGIR Workshop*.

[56] Christophe Van Gysel, Maarten de Rijke, and Marcel Worring. 2016. Unsupervised, Efficient and Semantic Expertise Retrieval. In *WWW*. ACM, 1069–1079.

[57] Christophe Van Gysel, Bart Goethals, and Maarten de Rijke. 2015. Determining the Presence of Political Parties in Social Circles. In *ICWSM*, Vol. 2015. 690–693.

[58] Nguyen Xuan Vinh, Julien Epps, and James Bailey. 2010. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *JMLR* 11 (2010), 2837–2854.

[59] Roger Weber, Hans-Jörg Schek, and Stephen Blott. 1998. A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In *VLDB*. 194–205.

[60] Yu Zhao, Liu Zhiyuan, and Maosong Sun. 2015. Representation learning for measuring entity relatedness with rich information. In *IJCAI*. 1412–1418.