Inducing Commonsense Knowledge Using Vector Space Embeddings

Zied Bouraoui

HABILITATION À DIRIGER DES RECHERCHES

Inducing Commonsense Knowledge Using Vector Space Embeddings

Zied Bouraoui
November 17th, 2022
Artois University

Jury:

Isabelle Bloch	Sorbonne Université
Jesse Davis	KU Leuven
Philippe Langlais	Université de Montréal
Daniel Le Berre	Université d'Artois
Henri Prade	IRIT Toulouse
Marie-Christine Rousset	Université Grenoble Alpes
Steven Schockaert	Cardiff University

CONTENTS

CHAPTER 1

INTRODUCTION

A note: This document aims to give a general overview of some of our work conducted between 2016 and 2022. To keep this document within a reasonable page limit, a high-level summary of the contributions is provided. For a more comprehensive discussion along with technical details and experimental results, the reader can refer to the original papers referenced along with the text and made available in the appendix. This document does not contain extensive related work sections, only some works are mentioned to help position our work. All the results were obtained together with my colleagues, and PhD Students, whom I would like to address my thanks: Steven Schockaert, Salem Benferhat, Rana Alshaikh, Na Li, Yixiao Wang, Shoaib Jameel, José Camacho-Collados, Luis Espinosa Anke, Víctor Gutiérrez-Basulto, Truong-Thanh Ma, Sébastien Konieczny, Ivan José Varzinczak, Nicolas Schwind, Ping Wang, Kun Yan, Qing Gu, Zied Loukil, Rym Mohamed, Faiz Gargouri, Chenbin Zhang, Jun Hou, and Shelan S. Jeawak, Pierre Marquis and Jean-Marie Lagniez. Finally, I would like to thank all my other coauthors with whom I have made several contributions, but not reported in this document.

1.1 Context and Research Questions

Commonsense knowledge plays an increasingly important role in the development of Artificial Intelligence (AI) systems. Knowledge can be expressed using different representation formalisms including symbolic frameworks, vector representations and textual descriptions, among others.

Symbolic knowledge representations. The traditional approach for encoding knowledge about concepts has been to use logic-based (symbolic) representations, typically in the form of a rule base. Such a rule base is often called an ontology in this context.

Example 1. *Consider the following rules:*

$$\mathsf{expertInAI}(X) \leftarrow \mathsf{authorOf}(X, Y), \mathsf{hasTopic}(Y, \mathsf{artificialIntelligence})$$
$$\mathsf{hasTopic}(X, \mathsf{artificialIntelligence}) \leftarrow \mathsf{hasTopic}(X, \mathsf{knowledgeRepresentation})$$
$$\mathsf{hasTopic}(X, \mathsf{artificialIntelligence}) \leftarrow \mathsf{hasTopic}(X, \mathsf{machineLearning})$$
$$\mathsf{hasTopic}(X, \mathsf{artificialIntelligence}) \leftarrow \mathsf{hasTopic}(X, \mathsf{multiAgentSystems})$$
$$\mathsf{hasTopic}(X, \mathsf{artificialIntelligence}) \leftarrow \mathsf{hasTopic}(X, \mathsf{naturalLanguageProcessing})$$

Here we have used the notational conventions from logic programming, where the conclusion of the rule is shown on the left-hand side and "," denotes conjunction. The first rule intuitively asserts that somebody who has published a paper on an AI topic is an expert in AI. The remaining rules encode that knowledge representation, machine learning, multi-agent systems and natural language processing are sub-fields of AI. Along with the ontology, we are usually given a set of facts, e.g.:

$$\{\mathsf{authorOf}(\mathsf{bob}, p), \mathsf{hasTopic}(p, \mathsf{knowledgeRepresentation})\}$$

Given the set of facts and the aforementioned rules, we can conclude that $\mathsf{hasTopic}(p, \mathsf{artificialIntelligence})$ *holds and thus also that* $\mathsf{expertInAI}(\mathsf{bob})$ *holds.*

Using ontologies for encoding conceptual knowledge has at least two key advantages. First, the formal semantics of the underlying logic ensures that knowledge can be encoded in a precise and unambiguous way. This, in turn, ensures that different applications can rely on a shared understanding of the meaning of the concepts involved. Second, ontologies enable us to capture knowledge in a transparent and interpretable way. Symbolic rules that have been learned from data can often be difficult to interpret, for instance, which makes it relatively straightforward to update knowledge and to support decisions with meaningful explanations. But ontologies, and symbolic approaches to knowledge representation more generally, also have important drawbacks. A first issue stems from the fact that the knowledge which is captured in an *ontology is rarely complete*. For instance, consider the following set of facts:

$$\{\mathsf{authorOf}(\mathsf{alice}, q), \mathsf{hasTopic}(q, \mathsf{planning})\}$$

As none of the available rules express that planning is a sub-field of AI, $\mathsf{expertInAI}(\mathsf{alice})$ can not be inferred. Nonetheless, to a human observer, it seems clear that this would be a valid inference, even without a precise understanding of what the predicate $\mathsf{expertInAI}$ is supposed to capture. Essentially, standard frameworks for modelling ontologies *lack a mechanism for inductive reasoning* [Gärdenfors, 2004]. This is not something which can be easily addressed, as inductive arguments rely on graded notions such as *similarity and typicality* [Rips, 1975; Osherson *et al.*, 1990; Sloman, 1993; Osta-Vélez and Gärdenfors, 2020]. Another issue is that many *concepts are difficult to characterise* in a satisfactory way using logical rules. For instance, somebody with a single published paper in AI would not normally be considered to be an AI expert, except perhaps if the work was particularly influential or groundbreaking, but formalising such notions using rules is challenging. Probabilistic [Gutiérrez-Basulto *et al.*, 2017; Borgwardt *et al.*, 2018] or Possibilistic extensions [Benferhat and Bouraoui, 2017; Mohamed *et al.*, 2018] of standard ontology languages may alleviate some of the aforementioned issues, but such frameworks still do not allow us to model similarity, or aspects that are a matter of degree (e.g. being an expert in AI).

Vector Space Representations. The most common alternative to ontologies is to encode conceptual knowledge using vector space representations. Most work on vector representations of conceptual knowledge has focused on Knowledge Graphs Embeddings (KGs) or Word Embeddigns (WEs). KGs are vector

space representations of the entities and relations that occur in a given set of relational triples of the form (e, r, f), where e and f are entities and r is a binary relation. The primary purpose of KGs is to identify plausible additional triples by modelling statistical dependencies among the considered relations. As an example, we may consider the following knowledge graph:

$$K = \{(\mathsf{bob}, \mathsf{authorOf}, \mathsf{p}), (p, \mathsf{hasTopic}, \mathsf{knowledgeRepresentation}), \\ (p, \mathsf{hasTopic}, \mathsf{artificialIntelligence}), (\mathsf{bob}, \mathsf{hasProperty}, \mathsf{expertInAI})\}$$

Approaches for Knowledge Graph Embedding (KGE) learn a vector representation $\mathbf{e} \in \mathbb{R}^n$ for each entity e and a scoring function $\phi_r : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ for each relation type r, such that $\phi_r(\mathbf{e}, \mathbf{f})$ captures the plausibility of the triple (e, r, f), i.e. the plausibility that the relation r holds between the entities e and f [Bordes *et al.*, 2013; Yang *et al.*, 2015; Trouillon *et al.*, 2017; Sun *et al.*, 2019]. The vector $\mathbf{e}$ is called the *embedding* of entity e. The purpose of KGE is at least two-fold. First, it is hoped that this embedding will uncover some of the underlying semantic dependencies in the KG, and that as a result, we will be able to identify plausible triples that are missing from the given KG. Second, by encoding the information that is captured in the knowledge graph using vectors, it becomes easier to exploit this information in neural network models. Figure 1.1 shows a vector encoding of the paper p and some of the considered subject areas. For this example, we assume that the dot product between p and a subject area indicates how relevant that subject area is to p, i.e. we have $\phi_{\mathsf{hasTopic}}(\mathbf{e}, \mathbf{f}) = \mathbf{e} \cdot \mathbf{f}$. Let us write $\mathbf{v_{ML}}$, $\mathbf{v_{AI}}$, $\mathbf{v_{NLP}}$ and $\mathbf{v_{KR}}$ for the vector representations of the different subject areas, and $\mathbf{p}$ for the representation of p. According to this embedding, we have $\mathbf{p} \cdot \mathbf{v_{ML}} \approx \mathbf{p} \cdot \mathbf{v_{NLP}} > \mathbf{p} \cdot \mathbf{v_{KR}}$, which captures the knowledge that p is more closely related to machine learning and natural language processing than to knowledge representation. Moreover, note how the norm of $\mathbf{v_{AI}}$ is larger than the norms of $\mathbf{v_{ML}}$, $\mathbf{v_{NLP}}$ and $\mathbf{v_{KR}}$. This intuitively captures the knowledge that the term artificial intelligence is broader in meaning. For instance, we can encode the knowledge that machine learning is a sub-discipline of AI by ensuring that for every vector $\mathbf{x} \in \mathbb{R}^2$ it holds that:

$$\mathbf{v_{ML}} \cdot \mathbf{x} < \mathbf{v_{AI}} \cdot \mathbf{x}$$

KGE models start from a structured knowledge graph and represent entities and relations as geometric objects in the learned embedding itself (e.g. as translations, linear maps, combinations of projections and translations, etc). In the absence of a KG, a common strategy consists in using a text corpus as input, where several knowledge extraction mechanisms could be used. For example, one can predict other instances of the *hasTopic* relation without explicitly being told that the relation of interest is *hasTopic* using relation extraction techniques. In such a case, sentences mentioning the example pairs are extracted from a large corpus and used to train some neural network models. To predict new instances of the relation, the resulting model can then be applied to other sentences from the given corpus. Another alternative is to use pre-trained Word Embeddings. Word embeddings aims to learn low-dimensional, dense and continuous vectors to represent the words ($\mathbf{v_{ML}}$, $\mathbf{v_{AI}}$, $\mathbf{v_{NLP}}$, etc). Word embedding encode both semantic and syntactic information, where semantic information mainly correlates with the meaning of words, while syntactic information refer to their structural roles. One of the most surprising aspects of word em-

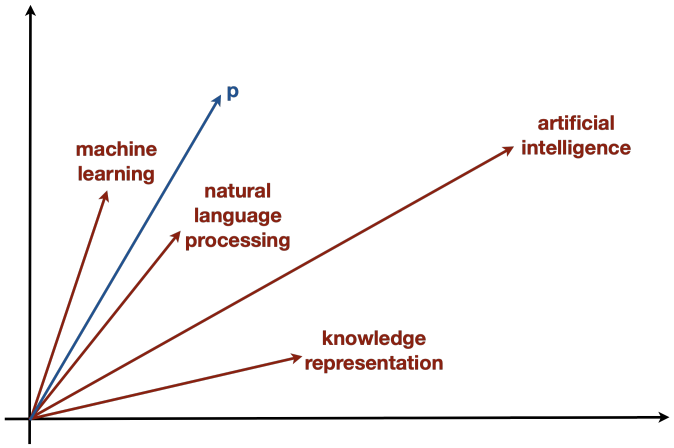


Figure 1.1: Illustration of a simple knowledge graph embedding, in which the dot product between p and a subject area indicates how relevant that subject area is to p.

beddings is the fact that they capture conceptual knowledge, despite essentially being trained to capture similarity. This is, for instance, illustrated in the fact that predicting analogical word pairs is a commonly used benchmark for evaluating word embeddings. Word embeddings can be divided into static embeddings and contextualised embeddings. Static vectors are context-free, while contextualised vectors are context-sensitive. One of the most popular methods to obtain static WEs is the Skip-gram (SG) model, as well as the related Continuous Bag Of Words (CBOW) model, which are often jointly referred to as word2vec [Mikolov *et al.*, 2013] and Glove [Pennington *et al.*, 2014]. Contextualised word embeddings, such as BERT [Devlin *et al.*, 2019], GPT-2 [Radford *et al.*, 2019a], and XLNet [Yang *et al.*, 2019], learn word vectors that are sensitive to the context in which they appear. They can capture many syntactic and semantic properties of words under diverse linguistic contexts. LMs capture prior knowledge about word meaning, and language more generally, in a powerful but opaque way.

Symbolic Vs vector space representation. When it comes to modelling conceptual knowledge, an important advantage of vector space representations is that they naturally support inductive inferences. Moreover, such representations are better suited for modelling graded notions such as similarity than symbolic representations. However, the extent to which "rule-like" knowledge can be captured is limited. As we saw in the aforementioned example, we can model the fact that one concept is subsumed by another, but it is not clear how more complex rules can be encoded using vector space embeddings. Moreover, embedding models lack the transparency of symbolic representations, which makes it harder to generate meaningful explanations or to update representations (e.g. to correct mistakes, add new knowledge, or take account of changes in the world). It is thus clear that symbolic knowledge (e.g. ontologies) and vector space embeddings (word embeddings or KG embeddings) have complementary strengths and weaknesses when it comes to modelling knowledge. This work aims to combine the best of the two worlds.

The available ontologies that encode such conceptual knowledge are inevitably incomplete and not

capable to model similarity, or aspects that are a matter of degree. As vector space representations provide the advantage to support support induction, a natural question we address is how to use these embeddings for automatically complete knowledge bases. The main problem with implementing such inductive strategies in practice comes from the fact that they often rely on types of background knowledge which is not usually available in symbolic form. As example [Dubois *et al.*, 1997; Schockaert and Prade, 2013], assume that we have the following knowledge about some concept C:

$$\mathsf{Strawberry} \sqsubseteq C \qquad \mathsf{Orange} \sqsubseteq C$$

Intuitively, even if we know nothing else about C, we could still make the following inductive inference:

$$\mathsf{Raspberry} \sqsubseteq C \tag{1.1}$$

This conclusion relies on background knowledge about strawberries, oranges and raspberries, in particular the fact that raspberries are expected to have all the *natural* properties that strawberries and oranges have in common (e.g. being high in vitamin C). The main aim to do induction is to determine which objects are likely to have some property P, knowing that the objects $o_1, ..., o_n$ have this property (but knowing nothing else about property P). In other words, inductive generalization in these models is based on our knowledge of the semantic features of the objects. This background knowledge can be obtained from vector space embeddings.

This work explores this problem in two directions. First, we focused on how to integrate vector space embeddings and ontologies for inductive reasoning with ontologies, with a focus on ontology completion. Second, we studied how to learn high-quality vector representations that can be used as background knowledge to support reasoning. The main idea is that vector space representations allow us to capture properties and similarity between different concepts and relations, which can be used to make plausible inferences. For instance, in the setting from Example 1, if we know that the vector representation of planning is highly similar to the vector representation of knowledgeRepresentation, we can plausibly infer the following rule:

$$\mathsf{hasTopic}(X, \mathsf{artificialIntelligence}) \leftarrow \mathsf{hasTopic}(X, \mathsf{planning})$$

While a number of embedding methods have recently been proposed to learn vector space representations, an important remaining problem is that they typically do not explicitly model concepts and relations. Namely, they only learn the representations of the objects (entities), while concept and relation representations are mostly important in for knowledge base completion. Accordingly, we study in this work how to model concepts in efficient way, in particular, in where only few examples are available for learning. Similarly, we address the question of performing relation induction in vector space embedding. With the emergence of contextualised language models, a natural question we addressed is whether we can distil meaningfully vectors that can be used as background knowledge that permit to support reasoning.

1.2 Summary of the contributions

The remainder of this document provide a brief summary of our work on:

Learning Conceptual Space Representations Modelling concepts as regions can supports the view that symbolic knowledge can be expressed as qualitative constraints on some underlying geometric model. This idea was developed in the 1990s by Gärdenfors in his theory of conceptual spaces [Gärdenfors, 2000]. The key characteristic of conceptual spaces is that concepts are represented as regions, rather than vectors. A rule of the form $A(x) \leftarrow B(x), C(x)$ can then be viewed as the constraint that the intersection of the regions representing B and C should be included in the region representing A. While the theory of conceptual spaces offers an elegant solution to combine symbolic and vector representations, in practice, it is often difficult to learn region-based representations of concepts from data. What matters in this context is (i) learning suitable entity embedding [Jameel *et al.*, 2017; Alshaikh *et al.*, 2020a] (ii) whether we can learn region-based representations of concepts [Bouraoui and Schockaert, 2018; Bouraoui *et al.*, 2020a], and (iii) whether we can learn vector representations in which dimensions are meaningful and organised into domains [Alshaikh *et al.*, 2020a, 2019, 2020b].

Modelling Relational knowledge Similarly to concepts, play important role in ontologies. A part of our work concerns modelling relations in vector spaces representations, with a focus on Word Embeddings. Based on the observation that that many relational knowledge can be modelled as vector translations in a embedding space, we first revisited this view by adding the fact that concepts can be model as regions on the space [Bouraoui *et al.*, 2017, 2018] and then considered another alternative aiming to characterize the relatedness between two entities by learning a relation vector in an unsupervised way from corpus statistics [Jameel *et al.*, 2018]. Finally, with the emergence of language models such as BERT, we studied how such language models capture more relational knowledge than standard embeddings, and in particular whether they can lead to improved performance on the relation induction tasks [Bouraoui *et al.*, 2020b].

Deriving High-Quality Vectors from Contextualised Language Models The ability to model word meaning *in context* is a central feature of transformer-based language models. Nonetheless, distilling static word vectors from language models is useful for several applications where word meaning has to be modelled in the absence of (sentence) context such as for ontology alignment, ontology completion, and zero-shot and few-shot learning. We first addressed the question of how to learn such representations from LMs [Li *et al.*, 2021; Wang *et al.*, 2021, 2022] and show the effectiveness of the learned vectors for few-shot image classification [Yan *et al.*, 2021a,b, 2022] and ontology completion [Li *et al.*, 2019].

Plausible Reasoning about Ontologies It is highly relevant for the development of robust AI systems to understand how symbolic approaches to AI can be made more flexible by equipping them with inductive capabilities, i.e. making it possible to infer likely concept inclusions (or rules) by using the

knowledge of the ontology in combination with the additional background knowledge provided by vector representations. In other words, one would like symbolic systems to incorporate mechanisms to use predictions made by neural approaches, informing about plausible situations witnessed in the data, in a principled way. We discuss ways in which this idea can be implemented form ontology completion [Bouraoui and Schockaert, 2019; Li *et al.*, 2019]. Using inductive mechanisms may introduce several conflicting knowledge and inconsistency. We introduced several methods for repairing knowledge bases Bouraoui *et al.* [2020d], in particular for ontology query answering [Mohamed *et al.*, 2018, 2022c,b]. We also studied how can traditional KR tasks such as ontology merging [Bouraoui *et al.*, 2020c, 2022b,a] can benefits from the conceptual spaces view.

CHAPTER 2

LEARNING CONCEPTUAL SPACE REPRESENTATIONS

Conceptual spaces were proposed by Gärdenfors as an intermediate representation level between symbolic and connectionist representations [Gärdenfors, 2000]. The theory of conceptual spaces has been extensively used in philosophy, e.g. to study metaphors and vagueness [Douven *et al.*, 2013], and in psychology, e.g. to study perception in domains such as color [Jäger, 2009]. Conceptual spaces support the view that symbolic knowledge can be expressed as qualitative constraints on some underlying geometric model. This chapter provides an overview of our work on learning concept space representations from data.

2.1 Conceptual Spaces

Conceptual spaces are geometric representations of knowledge, in which the objects from some domain of interest (e.g. movies) are represented as points in a metric space, and concepts (e.g. comedies) or properties (e.g. scary) are modelled as (possibly vague) convex regions. As such, they are similar in spirit to vector space representations that have been largely used in NLP and machine learning, but there are also notable differences. First, an explicit distinction is made between the entities from the domain of discourse, which are represented as vectors, and the corresponding properties and concepts, which are represented as regions. Second, a conceptual space is a high-dimensional vector space that is spanned by a set of **quality dimensions**. Each quality dimension represents a measurable and cognitively meaningful feature in the space, i.e. the quality dimension assigns a feature to the entities. This is illustrated in Figure 2.1, which shows a conceptual space of animals. Specific animals are represented as points in this space. Concepts such as mammal and properties such as scary are represented as regions. The dimensions capture the ordinal features dangerous and large. In this representation, the region modelling mammal is included in the region modelling vertebrate, which intuitively captures the rule $\mathsf{vertebrate}(X) \leftarrow \mathsf{mammal}(X)$, i.e. all mammals are vertebrates. Note how this representation can also capture semantic dependencies that are harder to encode using rules, e.g. the fact that large spiders are scary.

While it is convenient to think about conceptual spaces as vector space embeddings with some added structure, conceptual spaces do not necessarily have the structure of a vector space. A conceptual space is defined from a set of *quality dimensions* $Q_1, ..., Q_n$. Each of these quality dimensions captures a prim-

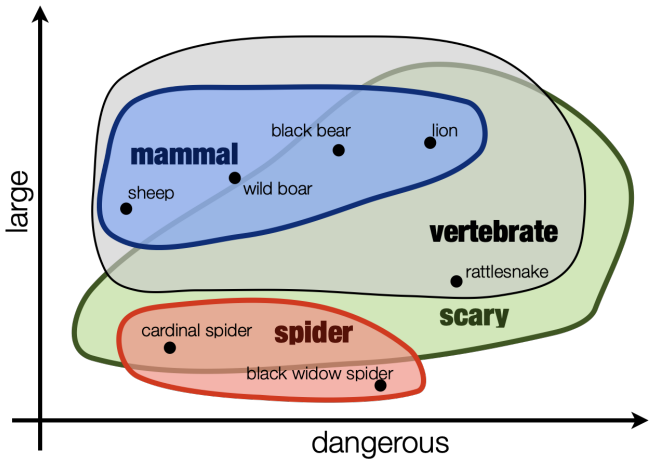


Figure 2.1: Illustration of a conceptual space of animals.

itive feature. We can then measure to what extent two entities are similar or different based on their distance with respect to that quality dimension. Features that describe temperature, weight, and height are examples of quality dimensions. The relation between two quality dimensions is either integral (if they inherently belong to the same aspect, e.g. *hue* and *brightness*), or separable (if they are meaningful in isolation from each, e.g. *size* and *brightness*). This distinction between integral and separable dimensions plays an important role in cognitive theories, as it affects how similarity is perceived. Based on a variety of psychological evidence, Euclidean distance is normally used when integral dimensions need to be combined, whereas Manhattan distance is used when separable dimensions need to be combined [Nosofsky, 1984; Gärdenfors, 2000]. Quality dimensions are partitioned into so-called *domains*, where dimensions that belong to the same domain are assumed to be integral, while dimensions from different domains are assumed to be separable. We can view domains as Cartesian products of quality dimensions. For instance, if D_i is composed of the quality dimensions $Q_1, ..., Q_k$ then the elements of D_i are tuples $(x_1, ..., x_k) \in Q_1 \times ... \times Q_k$. We can thus intuitively think of domains as vector spaces, although in general it is not required that domains satisfy the axioms of a vector space. An individual (e.g. a specific apple) is represented as an element $(x_1, ..., x_k)$ of a given domain, whereas we can think of properties (e.g. red, green, cold, warm) as regions. One of the central assumptions in the theory of conceptual spaces is that each *natural* property corresponds to a *convex* region in some domain. A *concept* is characterised in terms of a set of natural properties, along with information about how these properties are correlated. To define this notion of convexity, we have to assume that each domain D_i is equipped with a ternary betweenness relation $\mathsf{bet}_i \subseteq D_i \times D_i \times D_i$. A region $R \subseteq D_i$ is then said to be *convex* iff

$$\forall a, b, c \in D_i \,.\, a \in D_i \wedge c \in D_i \wedge \mathsf{bet}_i(a, b, c) \Rightarrow b \in D_i$$

In this case, we will only consider domains that correspond to Euclidean spaces, where the notion of convexity can be interpreted in the standard way. In the following, our focus will be first on learn entities representation that can be seen an approximation of conceptual spaces. Second, given an entity

embedding, we show how to learn region based representations of properties and concepts. Finally, we focus on identifying quality dimensions and grouping these quality-dimensions into domains.

2.2 Learning embedding as approximation of conceptual spaces

Let $\mathcal{E}$ be a set of entities, $\mathcal{S}$ a set of semantic types and $\mathcal{R}$ a set of relations. In [Jameel *et al.*, 2017], we proposed methods that aim to represent each entity e from $\mathcal{E}$ as a point $p_e \in \mathbb{R}^n$, such that all entities of the same semantic type (domain) belong to some lower-dimensional subspace. Those subspaces can be then seen as as approximations of conceptual spaces viewed as being themselves embedded in a more general vector space. In such embeddings, entities and by extension their properties have a natural geometric interpretation. The input of the model is a bag-of-words representation W_e of every entity e, a set of entities $\mathcal{E}_s$ for each semantic type $s \in \mathcal{S}$ and a set of knowledge graph triples $\mathcal{T}$ of the form (e, r, f), with $e, f \in \mathcal{E}$ and $r \in \mathcal{R}$. The entity embedding is learned by minimizing an objective function that contains three components: The first component (textual component), which is variant of GloVe model [Pennington *et al.*, 2014] for word embedding, that represents entities with similar bag-of-words representations using similar vectors. The second component (type component) aims to express the fact that each semantic type s is bounded by a set of points $q_1^s, ..., q_n^s$ such that every entity of that semantic type belongs the convex hull formed by $q_1^s, ..., q_n^s$. Finally, the third component (regularization component) add the constraint that the space spanned by $q_1^s, ..., q_n^s$ should be as low-dimensional as possible (for example, using nuclear norm regularization [Fazel, 2002]).

Motivated by the conceptual spaces view, we want to insure that all entities that have some properties (i.e. for which a given term is relevant) to be located in some well-defined region of the space. Intuitively, we want to impose that entities to which a given term applies should be separated from entities to which the term does not apply. Assuming that a term t applies to an entity e iff it occurs at least once in the bag-of-words representation W_e, we suggested in [Jameel *et al.*, 2017] as refinement of the textual component to add max-margin constraints that are derived from a bag-of-words representation of the entities. Using such constraints, we can jointly learn the vector representations of the entities p_e and hyperplane reflecting what information can be derived about the entity e from the fact that term t occurs in W_e. The max-margin constraints model provides then with an entity embedding that can be used to verifying which entities have a given property. However, many properties are a matter of degree. For example, modelling “tall mountains in France” or “influential music bands” require to model the extent to which each entity has the given property "tall" or "influential". As solution, in [Jameel *et al.*, 2017], we used the Positive Pairwise Mutual Information (PPMI) to estimate how much each term is related to each entity ($e \in \mathcal{E}, t \in W$). While there are other ways in which we can quantify how strongly a given term is related to an entity, PPMI is by far the most popular choice in the context of word embedding [Turney and Pantel, 2010] as it allows to measure the strength of association between a document d and a word w. Assuming that the bag-of-words representations of the entities contain the properties they have, we add the assumption that the more a property applies to an entity, the more it will be mentioned in textual descriptions of that entity. While this may seem like a strong assumption, it is indeed expected

that adjectives such as 'tall' will mostly be used in the context of the tallest mountains, thus allowing to distinguish exceptional elements (w.r.t. tallness). In addition, words such as 'snow', 'top, or 'cloud' which may appear more proportionally, allowing to differentiate between other mountains. To this end, we proposed a refinement of the textual component that relies on partitions of the entities according to terms using PPMI score. As shown in the experimental analysis we conducted in [Jameel *et al.*, 2017], the resulting entity embedding is interpretable, in the sense that we can use the model to describe what features a given entity has, or conversely to retrieve the entities that are most strongly related to a given set of query terms. For an example, if we consider the query "country population", using only the entity embedding, we obtain the following top-ranked entities: "china", "india", "america", "indonesia", "iran", "brazil". For "movies dinosaur" we obtain "dinosaur", "jurassicpark" as the top ranked entities, while for "france capital" we obtain "france", "capital", "paris".

While the methods we have proposed in [Jameel *et al.*, 2017] provide entity embeddings in which entities and their properties, have a natural geometric interpretation (e.g. subspaces to represent semantic types and hyperplanes to capture properties), an important remaining problem is that these methods typically do not explicitly model the concepts.

2.3 Modelling Concepts as Regions

In learned vector space embeddings, the objects (entities) from some domain of interest are represented as points or vectors, as in conceptual spaces. Most embedding models do not learn region-based representations of concepts. However, if we have access to a number of instances $c_1, ..., c_n$ of a given concept C, we can aim to learn a region-based representation of C from embeddings of these instances. The potential of this strategy stems from the fact that in many embedding models, these instances can be expected to appear clustered together in the vector space. To illustrate this, consider Figure 2.2, which shows the first two principal components of a 300-dimensional embedding of BabelNet concepts [Navigli and Ponzetto, 2012] using NASARI vectors[1], which have been learned from Wikipedia and are linked to BabelNet [Camacho-Collados *et al.*, 2016]. In the figure, the red points correspond to entities that are instances of the concept Artist, while the blue points correspond to entities that are instances of Painter. For instance, the embeddings of *Edouard Manet*, *Vanessa Bell* and *Claude Monet* appear close to the centre of the blue point cloud. As can be seen, painters appear as a distinct cluster in this vector space embedding. When attempting to learn a region-based concept representation, we are faced with two challenges: (i) we typically only have access to positive examples and (ii) the number of available instances is often much smaller than the number of dimensions in the vector space. This means that we inevitably have to make some simplifying assumptions to make learning possible.

Learning Gaussian Representations A natural choice is to represent concepts as Gaussians. This has the advantage that concept representations can be learned in a principled way, as the problem of

[1] Downloaded from `http://lcl.uniroma1.it/nasari/`.

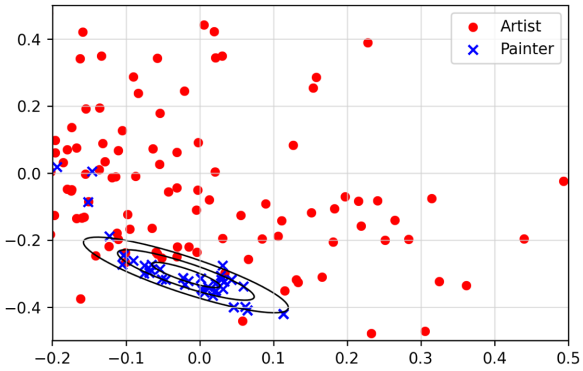


Figure 2.2: First two principal components of a vector space embedding of BabelNet entities, where blue points correspond to instances of the concept Artist and red points correspond to instances of the concept Painter, according to Wikidata.

estimating Gaussians from observations, either with or without prior knowledge, has been well-studied. Representing concepts using probability distributions, rather than hard regions, also fits well with the view that concept boundaries tend to be fuzzy and ill-defined more often than not. Note that in neural models, concepts are typically represented as vectors, with concept membership determined in terms of dot products, e.g. $\sigma(\mathbf{e} \cdot \mathbf{c})$ is often used to estimate the probability that the entity e (with embedding $\mathbf{e}$) is an instance of concept C (with embedding $\mathbf{c}$), with σ the sigmoid function. This choice effectively means that concepts are represented as spherical regions in the vector space. When using Gaussians, we relax this modelling choice, allowing concepts to be represented using ellipsoidal regions instead. To deal with the (typically) small number of instances that are available for learning a concept, in [Bouraoui *et al.*, 2017] we only considered Gaussians with diagonal covariance matrices. In this case, the problem simplifies to learning a number of univariate Gaussians, i.e. one per dimension. Moreover, a Bayesian formulation with a flat prior was used for estimating the Gaussians. As a consequence, concepts are actually represented using Student t-distributions. The practical implication is that slightly wider ellipsoidal regions are learned than those that would be obtained when using maximum likelihood estimates. Some contours of the learned distribution for the concept Painter are shown in Figure 2.2.

Bayesian learning with prior knowledge As mentioned above, in [Bouraoui *et al.*, 2017] we used a Bayesian formulation for learning Gaussian concept representations. While a flat (i.e. non-informative) prior was used in that paper, the same formulation can be used with informative priors, which offers a natural strategy for incorporating prior knowledge about the concept C being modelled. Such prior knowledge is particularly important when the number of available instances of C is very small (or, in an extreme case, when no instances of C are given at all). This idea was developed in [Bouraoui and Schockaert, 2018], where we consider two sources of prior knowledge were used: ontologies and vector space embeddings of the concept names. In both cases, the prior knowledge allows us to relate the target concept C to other concepts. However, in practice we typically do not yet have a representation of these

other concepts, i.e. we are trying to jointly learn a representation of all concepts of interest. This creates circular dependencies, e.g. the representation of concept A provides us with a prior on the representation of concept B, but the representation of concept B also provides us with a prior on the representation of A. This can be addressed using Gibbs sampling as we explained in [Bouraoui and Schockaert, 2018].

Priors on Mean. Suppose we have concept inclusions of the form $(C \sqsubseteq D_1)$,...,$(C \sqsubseteq D_k)$, and suppose we have a Gaussian representation of the concepts $D_1, ..., D_k$. Then we can induce a prior on the mean of the Gaussian representing C based on the idea that the mean of C should have a high probability in the Gaussians modelling $D_1, ..., D_k$. This can be achieved efficiently by taking advantage of the fact that the product of k Gaussians is proportional to another Gaussian. In addition to ontologies, we can also use vector space embeddings of the (names of the) concepts $C, D_1, ..., D_k$. Specifically, in [Bouraoui and Schockaert, 2018] we proposed a strategy based on the view that there should be a fixed vector offset between the embedding of a concept C and the mean of the Gaussian that represents it.

Priors on Variance. To obtain a prior on the variance of the Gaussian representing C, we take the view that this variance should be similar to that of the concepts that are most similar to C. To find such concepts, we could take the siblings of C in an ontology, the concepts whose vector space embedding is most similar to the embedding of C, or we could use a hybrid strategy where we select the siblings whose embedding is most similar to that of C. We again refer to [Bouraoui and Schockaert, 2018] for details.

Exploiting contrast sets A common strategy for learning conceptual space representations is to associate each concept with a single point, which intuitively represents its prototype [Gärdenfors and Williams, 2001]. The region representing a given concept C then consists of all points that are closer to the prototype of C than to the prototype of any other concept, i.e. concept regions are obtained as the Voronoi tessellation of a set of prototype points. This strategy is appealing, because it allows us to learn concept regions with a much wider extension than when learning Gaussians, especially in cases where we only have a few instances per concept. The main idea is illustrated in Figure 2.3, where we are interested in learning a region for the concept C. When using Gaussians, we would end up with ellipsoidal regions (contours) similar to the ones displayed in the figure. As a result, most points of the space are not assigned to any of the concepts. In contrast, if we construct prototypes by averaging the embeddings of the instances of a concept, and compute the resulting Voronoi tessellation, we essentially carve up the space, as also illustrated in the figure. To see why this can be beneficial in practice, Figure 2.4 shows the vector representations of the instances of three concepts: Songbook, Brochure and Guidebook. Now consider the left-most test instance of Songbook. If we are only given the training instances of this concept, this test instance is unlikely to be covered by the resulting representation. In contrast, if we instead attempt to carve up the space into regions corresponding to Songbook, Brochure and Guidebook, then this test instance would be classified correctly. The problem with implementing the aforementioned idea is that it only works if we are given a set of concepts that form a *contrast set* [Goldstone, 1996], i.e. a set of mutually exclusive natural categories that exhaustively cover some sub-domain. For example,

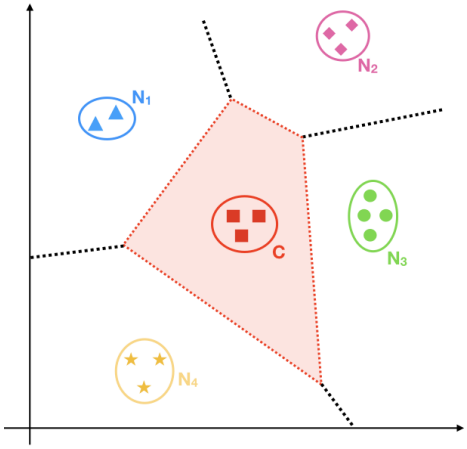


Figure 2.3: Estimating concept regions based on conceptual neighbourhood.

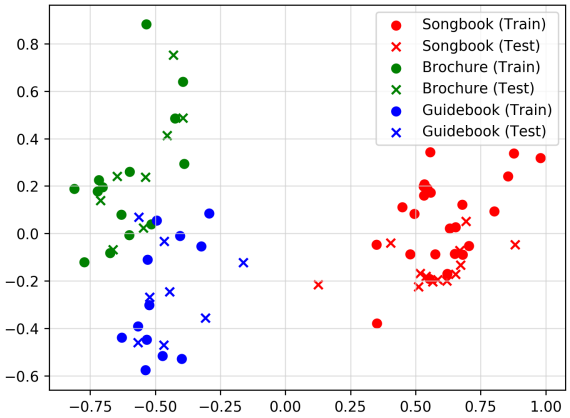


Figure 2.4: Instances of three BabelNet categories which intuitively can be seen as conceptual neighbors. The figure shows the first two principal components of the NASARI vectors.

the set of all common color names, the set $\{\mathsf{Fruit}, \mathsf{Vegetable}\}$ and the set $\{\mathsf{NLP}, \mathsf{IR}, \mathsf{ML}\}$ can intuitively be thought of as contrast sets. We say that two concepts are conceptual neighbours if they belong to the same contrast set and compete for coverage (i.e. are adjacent in the resulting Voronoi tessellation).

Existing ontologies do not typically describe contrast sets or conceptual neighbourhood. To deal with this, in [Bouraoui *et al.*, 2020a] we introduced a strategy for learning conceptual neighbourhood from data, i.e. for discovering pairs of concepts that are conceptual neighbours. Note that we have focused on conceptual neighbourhood rather than contrast sets, as the need for contrast sets to be exhaustive is difficult to guarantee. The method then relies on the simplifying assumption that the target concept C, along with its known conceptual neighbours $N_1, ..., N_k$ forms a contrast set. To represent the concept C, first a Gaussian is learned by pooling the instances of $C, N_1, ..., N_k$ together. The ellipsoidal contours of this Gaussian are then carved up into sub-regions for $C, N_1, ..., N_k$ by learning logistic regression classifiers. Specifically, the region representing C is obtained by training logistic regression classifiers

that separate the instances of C and N_i, for each $i \in \{1, ..., k\}$. To learn conceptual neighbourhood from data, the first step of the strategy from [Bouraoui *et al.*, 2020a] consists in generating weakly supervised training examples. To this end, we start with two concepts A and B that are siblings in a given taxonomy (i.e. concepts that have the same parent) and for which a sufficiently large number of instances is given. We then compare the performance of the following two types of concept representations:

1. Learn a Gaussian representation of A and B from their given instances.
2. Learn a Gaussian representation from the combined instances of A and B, and then split the resulting region by training a logistic regression classifier that separates A-instances from B-instances.

If the second representations perform (much) better at classifying held-out instances, we can assume that A and B are conceptual neighbours. If the second representations perform much worse, then we can assume that A and B are not conceptual neighbours. In case the performance is similar, then the pair A, B is disregarded when constructing the weakly labelled training set. Table 2.1 shows some examples of pairs of concepts A, B that were predicted to be conceptual neighbours using this process. Given the resulting training set, we can then train a standard text classifier on sentences that mention both A and B from some text corpus. Consider, for instance, the concepts *Hamlet* and *Village*, and the following sentence [2]:

> *In British geography, a hamlet is considered smaller than a village and ...*

The sentence suggests that *hamlet* and *village* are conceptual neighbors as it makes clear that these concepts are closely related but different. Once a classifier is trained, based on the weakly supervised training set, we can then apply it to other concepts. To learn the representation of a given target concept C (e.g. a concept with only few known instances), we can then use the text classifier to identify which of its siblings, in a given taxonomy, are most likely to be conceptual neighbours, and determine the representation of C accordingly. Tables 2.2 and 2.3 show some examples of the top conceptual neighbor predicted by the text classifier, for different target concepts. In particular, Table 2.3 shows examples where the resulting concept representations (i.e. the representations of the target concepts obtained by exploiting the predicted conceptual neighbourhood) were of high quality, as measured in terms of F1 score for held-out entities. Similarly, Table 2.2 shows examples where the resulting concept representations were of low quality. As can be seen, the predicted conceptual neighbours in Table 2.3 are clearly more meaningful than the predicted neighbours in Table 2.2. This illustrates how the quality of the concept representations is closely linked to our ability to find meaningful conceptual neighbours. Overall, the experiments in [Bouraoui *et al.*, 2020a] showed that using predicted conceptual neighbourhood, on average, led to much better concept representations than when estimating Gaussians from the known instances of the target concept.

[2] https://en.wikipedia.org/wiki/Hamlet_(place)

High confidence	Medium confidence
Actor – Comedian	Cruise ship – Ocean liner
Journal – Newspaper	Synagogue – Temple
Club – Company	Mountain range – Ridge
Novel – Short story	Child – Man
Tutor – Professor	Monastery – Palace
Museum – Public aquarium	Fairy tale – Short story
Lake – River	Guitarist – Harpsichordist

Table 2.1: Selected examples of siblings A–B which are predicted to be conceptual neighbours with high and medium confidence.

Concept	Top neighbor	F1
Bachelor's degree	Undergraduate degree	34
Episodic video game	Multiplayer gamer	34
501(c) organization	Not-for-profit arts organization	29
Heavy bomber	Triplane	41
Ministry	United States government	33

Table 2.2: Top conceptual neighbors selected for categories associated with a low F1 score.

2.4 Learning Quality Dimensions

The dimensions of learned vector spaces do not normally correspond to semantically meaningful properties. This is an important difference with conceptual spaces, which severely limits the interpretability of learned vector space representations. In section 2.2, we presented our work on learning interpretable embeddings. In this section, we review our work that has focused on mitigating this issue, by identifying interpretable directions in pre-trained vector spaces. These interpretable directions can then play the role of quality dimensions.

Identifying quality dimensions Assume that a set of entities $\mathcal{E}$ is given, together with a vector space embedding $\mathbf{e} \in \mathbb{R}^n$ for each entity $e \in \mathcal{E}$. To find interpretable directions, we can used the method from Derrac and Schockaert [2015] to learn interpretable dimensions that are similar to quality dimensions of the conceptual space. The strategy from [Derrac and Schockaert, 2015] relies on the assumption that a text description D_e is available for each entity e. Let W be the set of all words (or common multi-word expressions such as "New York") that appear in these descriptions D_e. For $w \in W$, we say that the word w is relevant for the entity e if w appears at least once in the description D_e. Then, the method from [Derrac and Schockaert, 2015] consists in learning a linear classifier in the embedding space for each $w \in W$ that separates the entities for which w is relevant from those for which this is not the case. The words $w_1, ..., w_n$ for which this classifier performs sufficiently well are then considered as semantic features. Each of these basic features w is then associated with a corresponding vector $\mathbf{w}$ (i.e. the normal vector of the separating hyperplane that is learned by the classifier). These candidate vectors are then clustered, and each cluster is treated as a quality dimension. This clustering step has the advantage that quality dimensions become easier to interpret, as we have a set of words to describe them,

Concept	Top neighbor	F1
Amphitheater	Velodrome	67
Proxy server	Application server	61
Ketch	Cutter	74
Quintet	Brass band	67
Sand dune	Drumlin	71

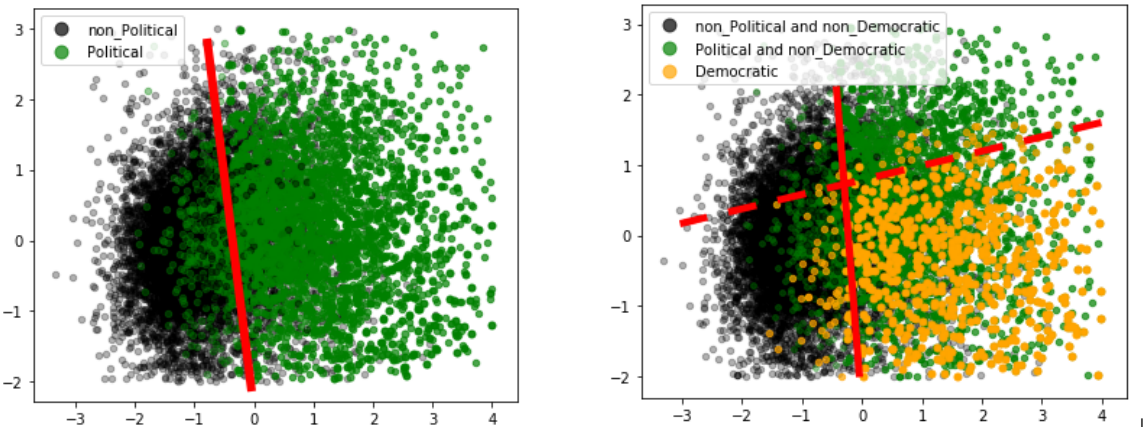
Table 2.3: Top conceptual neighbors selected for categories associated with a high F1 score.

rather than a single word, and it ensures that different quality dimensions are sufficiently different.

However, many semantic features do not make sense for all entities. For instance, in an embedding of movies, we may consider a feature that captures how closely a movie adheres to the book it is based on. This feature is relevant for book adaptation movies, but it is non-sensical for other movies. As an important practical implication, if quality dimensions are learned from the full set of entities, while only being sensical for a subset of these entities, we may expect them to be sub-optimal. Figure 2.5 illustrates this problem. It displays a projection of an embedding of organisations where the green dots in Figure 2.5a, correspond to those whose associated description contains words such as *political, politic, party, parties, politicians*. The method from Derrac and Schockaert [2015] discovered this cluster as a semantic feature. Now consider Fig. 2.5b, where the yellow dots correspond to organisations whose descriptions contain words such as *democratic* and *left-wings*. While this cluster describes a feature that is intuitively clear (i.e. organisations associated with left-wing political ideas), this feature is only relevant for a subset of organisations (i.e. political ones). A key, and perhaps surprising, observation is that this is reflected in the vector space. In particular, as can be seen in the figure, this feature cannot be characterized well using a single hyperplane. As solution, one can decompose the embedding into different domains. However, finding a suitable decomposition is a highly non-trivial problem, especially in unsupervised settings.

Instead of trying to find a hard decomposition of the entity embedding into separate domains, in [Alshaikh *et al.*, 2020b], we proposed a method based on the application of the method from [Derrac and Schockaert, 2015] in a hierarchical fashion. As results, in the example from Figure 2.5, we can view the feature *political* as defining a domain. To obtain a suitable characterization of the feature *democratic*, it then suffices to apply the method from [Derrac and Schockaert, 2015] to the *political* domain instead of to the full space, i.e. to the subset of entities that are considered to have the feature *political* to a sufficient extent. As confirmed by the experimental results that we conducted in [Alshaikh *et al.*, 2020b], learning the features in such a hierarchical way leads to semantically more meaningful representations. We showed in particular that a hierarchical relationship that exists between features (e.g. the fact that *democratic* is a sub-feature of *political*) is effectively reflected in the structure of the entity embedding.

Organising quality dimensions into domains The quality dimensions of a conceptual space are organised into domains. Accordingly, as we have seen in the previous section, the quality dimensions that can be identified in learned vector spaces also intuitively belong to different kinds. It would be of interest to group quality dimensions of the same kind together, to learn a structure which is akin to



(a) (b)

Figure 2.5: Projection of a 100-dimensional embedding of organisations, showing (a) how organisations that are described with words such as *political, politics,party,parties,politicians* (shown in in green) are separated from others; and (b) how organisations that are described using words such as *democratic, left-wings* (in yellow) are separated from others.

conceptual space domains. For instance, in the movies domain, we could imagine one group of quality dimensions about the emotion a movie evokes, as well as groups about the genre, the cinematographic style, etc. We will refer to these groups of learned quality dimensions as *facets*, rather than domains, to avoid confusion (e.g. domain can also refer to the domain-of-discourse, such as movies, or to the domain of a description logic interpretation) and to highlight the fact that there are still important differences between these facets and conceptual space domains. In addition to grouping quality dimensions that are concerned with the same aspect of meaning, we also want to learn a corresponding lower-dimensional vector space for each facet. In other words, the central aim is to decompose the given vector space into a number of lower-dimensional spaces, each of which captures a different aspect of meaning.

Note that we cannot learn these facets by simply clustering the quality dimensions. For instance, *thriller* and *scary* may be represented by similar directions in the vector space, but they should be assigned to different facets. In contrast, *romance* and *horror* would be represented by dissimilar directions but nonetheless belong to the same facet. The key solution, which we developed in [Alshaikh *et al.*, 2019] and [Alshaikh *et al.*, 2020a], is to rely on word embeddings to identify words that describe properties of the same kind. For instance, the word embeddings of different movie genres tend to be similar, because such words tend to appear in similar contexts. In the same way, different adjectives describing emotions tend to be represented using similar word vectors. This suggests a simple strategy for learning facets: (i) cluster the word vectors of the words associated with the quality dimensions that were identified in the given vector space; and (ii) represent the facet by the vector space that is spanned by the quality dimensions that are assigned to it. Unfortunately, this strategy was found to perform poorly in [Alshaikh *et al.*, 2019]. The main reason is that in many areas there is one dominant facet, such as the genre in the case of movies. When applying the aforementioned strategy, what happens is that each of the resulting facet-specific vector spaces mostly models the dominant facet. To address this issue, we

proposed in [Alshaikh *et al.*, 2019] an iterative strategy, in which the dominant facet is first identified and then explicitly disregarded when determining the second facet, etc. Another practical challenge is that the overall method is computationally demanding, especially the fact that a linear classifier has to be learned for each word from the vocabulary, to identify the interpretable directions (in the overall space and in each of the lower-dimensional facet-specific spaces). To address this issue, in [Alshaikh *et al.*, 2020a] we introduced a model that directly learns facet-specific vector spaces from bag-of-words representations of the entities, using a mixture-of-experts model to generalize the GloVe [Pennington *et al.*, 2014] word embedding model. Using this approach, facet-specific vector spaces can be learned much more efficiently, and moreover the resulting embeddings tend to be of a higher quality. The main limitation, however, is that this model assumes that suitable vector spaces can be learned from bag-of-words representations (rather than being agnostic to how the initial vector space embedding is learned) and that GloVe is a suitable embedding model for learning these vector spaces.

The resulting facet-specific embeddings can be used in a number of different ways. Perhaps the most immediate application of such representations is that they facilitate concept learning. For instance, suppose we want to represent each concept as a Gaussian. Furthermore, suppose that only one of the facet-specific vector spaces is relevant for modelling the considered concept. If we learn a Gaussian in each of the factor-specific vector spaces, we should end up with Gaussian with a large variance for the irrelevant facets, and a Gaussian with a much lower variance in the vector space corresponding to the relevant facet. This advantage of facet-specific vector spaces was empirically confirmed in [Alshaikh *et al.*, 2020a]. Moreover, they found that even strategies that only rely on the resulting quality dimensions, e.g. learning low-depth decision trees, were benefiting from learning facet-specific vector spaces, as the lower-dimensional nature of each vector space acts as a regulariser.

2.5 Conclusion

In this chapter, we discussed a number of strategies that are inspired by the theory of conceptual spaces. We looked at the possibility of combining symbolic and vector representations based on the idea that concepts can be viewed as regions in vector space embeddings. Moreover, we also explored the idea that meaningful "quality dimensions" can be identified in learned embeddings, adding more structure and a degree of interpretability to the vector representations themselves.

CHAPTER 3

MODELLING RELATIONAL KNOWLEDGE

Despite essentially being trained to capture word similarity, one of the most surprising aspects of word embeddings, such as those learned using Skip-gram and GloVe, is the fact that they capture relational knowledge. For example, word embeddings have been shown useful to complete analogy questions of the form a:b::c:?, asking for a word that relates to c in the same way that b relates to a, by predicting the word w that maximizes the following function $\cos(\mathbf{b} - \mathbf{a} + \mathbf{c}, \mathbf{w})$. In this chapter, we explore how relational knowledge can be modelled in word embeddings. We first provide an overview of our work on relation induction using embedding, which is the problem of predicting likely instances of a given relation based on some example instances of that relation. For instance, given the example pairs (*paris*, *france*), (*tokyo*, *japan*), (*Brussels*, *Belgium*), a relation induction system should predict other instances of the capital-of relation without explicitly being told that the relation of interest is the capital-of relation. With the success of language models, such as BERT [Devlin *et al.*, 2019], GPT2 [Radford *et al.*, 2019b] and XLNet [Yang *et al.*, 2019], we studied to what extent these LMs capture meaningful attributional and relational knowledge. Finally, rather than aiming to derive relational information from word embeddings, we studied how can learn relation vectors from distributional statistics, i.e. vectors encoding the relationship between two words.

3.1 Relation Induction in Word Embeddings Revisited

Given a set $\{(s_1, t_1), ..., (s_n, t_n)\}$ of word pairs that are related in a given way, where s and t as the source and target word respectively and $\mathbf{w}$ for the vector representation of the word w. The relation induction task aims to predict which other word pairs (s, t) are likely to be related in the same way. Following the observation that many lexical relationships can be modelled as vector translations in a word embedding [Mikolov *et al.*, 2013; Pennington *et al.*, 2014], one natural way to model a relation is to consider the average of translation vector $\mathbf{r} = \frac{1}{n}\sum_i(\mathbf{t_i} - \mathbf{s_i})$, and accept (s, t) as plausible instance if $\cos(\mathbf{s}+\mathbf{r}, \mathbf{t})$ is sufficiently high. While this approach was found efficient for modelling analogies [Drozd *et al.*, 2016], we found out in [Bouraoui *et al.*, 2018] that it leads to too many false positives when it come to relation induction. This is illustrated in Table 3.1 for the case where the vector translation $\mathbf{r}$ is constructed from the instances of the *capital of* relation of the BATS dataset [Gladkova *et al.*, 2016]. As can be seen in Table 3.1, most of the top-ranked pairs are actually incorrect. In practice, we identified two problems: the first problem is related to class imbalance and the other to the use of cosine similarity as a way to predict likely instances. More precisely, for relation induction task, the number of negative pairs

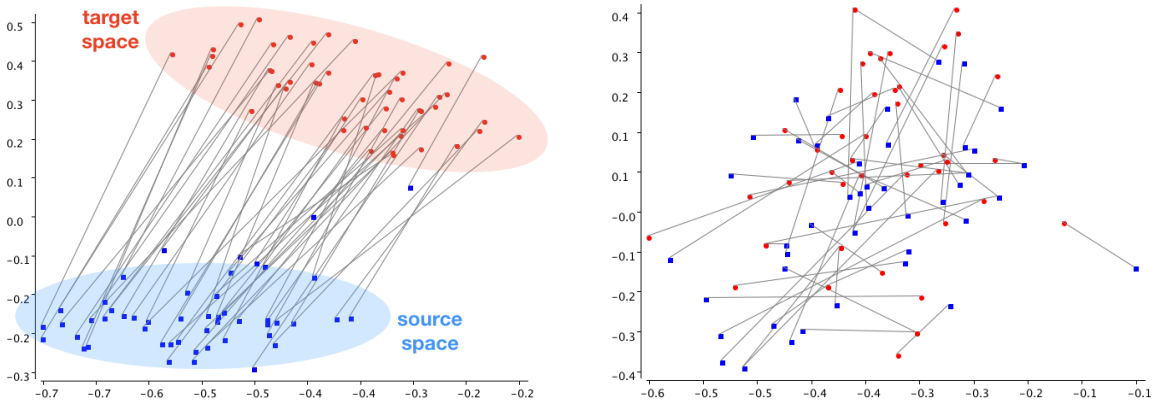
word pair	cos	word pair	cos
(horse, horses)	0.84	*(baghdad,iraq)*	0.64
(boy, girl)	0.79	(aware, unaware)	0.64
(madrid, spain)	0.73	*(moscow, russia)*	0.63
(london, england)	0.69	*(berlin, germany)*	0.63
(spain, madrid)	0.68	(look, looking)	0.61
(walk, walks)	0.65	(moscow,germany)	0.59

Table 3.1: Cosine scores for the average translation model applied to the *capital of* relation; correct instances are shown in italics.

is typically much higher than the number of positive pairs. As consequence, even the correct instances may get higher scores in general, due to the imbalance many incorrect instances will also receive very high scores. This is also due to the use of cosine similarity which treats all dimensions in the same way when comparing the vectors $\mathbf{r}$ and $\mathbf{t} - \mathbf{s}$. In practice, however, some dimensions of the word embedding may correspond to features of meaning that are irrelevant for the considered relationship. Note that for the analogy completion task, the cosine similarity is a suitable choice as we are only given one example (s, t) of a correct instance from which one cannot determine which dimensions are most relevant. For relation induction, however, we can use the empirical variance of the translation vectors $\mathbf{t_i} - \mathbf{s_i}$ to make a more informed choice.

Translation Model. To tackle these problems, in [Bouraoui *et al.*, 2018] we proposed a probabilistic relation induction model based on distributions over words that can occur as the first and second argument in valid instances and distributions over vector translations. This model has two main advantages. First, motivated by the view that concepts can be modelled as soft region in a space (as described in the previous chapter), the model learns a soft constraint on which words are likely to occur as source and which words are likely to occur as target words. This allows to reduce the number of spurious instances that are detected. Second, the model uses a Bayesian estimation of a Gaussian distribution over translation vectors to encode which features of word meaning are most important for the considered relation. Figure 3.1a gives an illustration of the model. Irrespective of the translation between source and target word, we would expect that (s, t) is a valid relation instance if the source $\mathbf{s}$ belongs to the same subspace as $\mathbf{s_1}, ..., \mathbf{s_n}$, and similar for the target word. Imposing this condition intuitively allows to ensure that only pairs where s and t are of the correct type are considered, which allows us in turn to substantially reduce the number of false positives that are predicted by the model. While translation vectors $\mathbf{t_i} - \mathbf{s_i}$ are all rather similar, as we can see in Figure 3.1a, there is no single translation vector that perfectly models the relation. To this end, we consider in addition a probability distribution over vector translations. In the example of Figure 3.1a, this distribution would have a small variance along directions which are orthogonal to the average translation vector (as most vectors are almost parallel), but a larger variance along the direction of the average translation vector itself (as the translation vectors have varying lengths). Putting everything together, the model we proposed in [Bouraoui *et al.*, 2018] accepts (s, t) as a valid instance if (i) $\mathbf{s}$ and $\mathbf{t}$ are sufficiently similar to the vector representations of the given source and target words, and (ii) the translation $\mathbf{t} - \mathbf{s}$ has a sufficiently high probability.

Regression Model. While there are several relations that can be approximately modelled as vector



(a) Superlative degree relation.

(b) Meronymy relation.

Figure 3.1: Two first principle components of the set $\{\mathbf{s_1}, ..., \mathbf{s_n}, \mathbf{t_1}, ..., \mathbf{t_n}\}$ of Superlative degree and Meronymy relations. The word vectors are from pre-trained Skip-gram embedding from the Google news corpus.

translations, there are many other relations for which this is not the case. Figure 3.1b illustrates this problem. It displays instances from the meronymy relation from the DiffVec dataset [Gladkova *et al.*, 2016]. We can clearly see that for the considered word embedding, a translation-based model will lead to sub optimal results. As an alternative, in [Bouraoui *et al.*, 2018] we proposed a Bayesian linear regression that weakens the translation assumption, and merely assumes that there is a linear mapping from source to target words. While it can potentially be more faithful, in practice, learning such a model requires a larger number of training instances to be effective. In fact, while a translation can be estimated from a single example, one can only learn a linear mapping if the number of training examples is higher than the number of dimensions. We addressed this problem in [Bouraoui *et al.*, 2018] by reducing the number of dimensions of the source space, based on the available number of training examples. Overall, the experimental evaluation in [Bouraoui *et al.*, 2018] showed that the regression model is outperformed by the translation model on average, except for the relations where the number of examples is rather large, e.g. "Event", "Hyper" and "Mero" relation from DiffVec (resp. 3583, 1173, 2825 instances).

3.2 Relation Induction using Language Models

Recently, the use of pre-trained contextualized language models (CLMs) such as BERT [Devlin *et al.*, 2019], RoBERTa [Liu *et al.*, 2019], GPT2 [Radford *et al.*, 2019b] and XLNet [Yang *et al.*, 2019] has led to substantial performance increases in a variety of NLP tasks. A natural question is thus whether such language models capture more relational knowledge than standard word embeddings, and in particular whether they can lead to improved performance on the relation induction task. CLMs are trained to complete sentences containing blanks, e.g. MaskedLMs. To address this question, one natural approach to see whether we can extract some relational knowledge from these LMs is to consider sentences that express a relation. Table 3.2 shows some BERT predictions for a number of different sentences that we

Sentence	BERT
The color of the banana is ___.	yellow
The color of the avocado is ___.	yellow
The color of the carrot is ___.	yellow
The color of the tomato is ___.	white
The color of the kiwi is ___.	white
The capital of Japan is ___.	tokyo
The capital of France is ___.	paris
The capital of Australia is ___.	canberra
The capital of the US is ___.	washington
The capital of Brazil is ___.	santos
Recessions are caused by ___.	inflation
Recessions are often caused by ___.	stress
Hangovers are caused by ___.	stress
I took my umbrella because it was ___.	warm
He didn't go to school because it was a ___.	secret
I like to have ___ for breakfast.	them
Her favorite subject in school was ___.	english
His favorite day of the week is ___.	christmas
They saw lots of scary animals such as ___.	bears
He likes ___ and most other vegetables.	potatoes

Table 3.2: Predictions by the BERT-Large-Uncased pre-trained language model for selected sentences.

hand craft to probe LMs. As can be seen, the performance of BERT is rather mixed. While it seems to have learned the capital-of relation well (notwithstanding the incorrect prediction for Brazil), BERT does not seem capable to capture color properties, as it predicts either *yellow* or *white* for all examples. Moreover, the most important insight from Table 3.2 comes from the two sentences about the cause of recessions, where the addition of the word *often* makes a difference between a sensible prediction (*inflation*) and a meaningless one (*stress*). This suggests that even if language models capture relational knowledge, it is important to find the right sentence triggers to extract that knowledge. In [Bouraoui *et al.*, 2020b] we proposed a methodology for distilling relational knowledge from a pre-trained language model. Starting from a few seed instances of a given relation, we first find trigger sentences that are likely to express that relation based on a large text corpus. These extracted sentences are then used as templates to fine-tune a language model to predict whether a given word pair is likely to be an instance of some relation, when given an instantiated template for that relation as input.

Let us write $\phi(s, t)$ to denote a sentence which mentions some source (head) word s and target (tail) word t, e.g.:

$$\phi(Paris, France) = \underline{Paris}\ is\ the\ capital\ of\ \underline{France}.$$

Such sentences are then treated as templates, which can be instantiated with different word pairs, e.g.:

$$\phi(Rome, Italy) = \underline{Rome}\ is\ the\ capital\ of\ \underline{Italy}.$$
$$\phi(Rome, France) = \underline{Rome}\ is\ the\ capital\ of\ \underline{France}.$$
$$\phi(Trump, Obama) = \underline{Trump}\ is\ the\ capital\ of\ \underline{Obama}.$$

A LM is then used to determine whether the resulting sentences formed by instantiating the templates with a new word pair (s, t) remains natural. If so, then (s, t) is likely to be an instance of the same relation. Intuitively, we expect that the LM should be able to distinguish between a natural sentence such as $\phi(\textit{Rome}, \textit{Italy})$ and unnatural sentences such as $\phi(\textit{Rome}, \textit{France})$ and $\phi(\textit{Trump}, \textit{Obama})$. Note that the text corpus is only considered to find predictive trigger sentences and the prediction about the pair (s, t) does not rely on any sentences from the corpus. This means that the accuracy of the predictions purely relies on the relational knowledge that is captured in the pre-trained LM. However, the example mentioned above relies on the fact that the template ϕ is indicative of the capital-of relation. This is not the case for all sentences mentioning *Paris* and *France*. For instance, consider the following sentence:

$$\phi'(\textit{Paris}, \textit{France}) = \textit{The Eiffel tower is in Paris, France.}$$

A sentence such as $\phi'(\textit{Rome}, \textit{Italy})$ is obliviously unnatural as ϕ' cannot be used to find new instances of the capital-of relation. In [Bouraoui *et al.*, 2020b], we proposed a method to filter out the templates that are not natural and only selecting those which are such that most of the sentences when instantiated are natural. The selected sentences express the considered relationship in general, rather than being specifically about a particular word pair. For many of the extracted sentences this may not be the case, as they might simply mention the two words for an unrelated reason (e.g. "Paris Hilton arrived in France today.") or they might only be sensible for the particular word pair (e.g. "The Eiffel Tower is located in Paris, France."). However, one can find some sentences that may not directly express the considered relation, but might nonetheless provide some useful evidence about it. Consider for instance the following sentences:

$$\phi_1 : \quad \textit{Paris is located in central France.} \tag{3.1}$$

$$\phi_2 : \quad \textit{Paris is the largest city in France.} \tag{3.2}$$

$$\phi_3 : \quad \textit{Paris is one of the oldest cities in France.} \tag{3.3}$$

While none of these sentences asserts the capital-of relationship, a word pair (s, t) for which the assertions $\phi_1(s, t)$, $\phi_2(s, t)$ and $\phi_3(s, t)$ are all true is nonetheless likely to be an instance of the capital-of relation. We proposed a way to rank the templates $\phi_1, ..., \phi_m$ by their usefulness keeping templates that directly express the relation (high score) and those providing indirect evidence. Table 6 shows five templates which were obtained for the *currency* and *capital-of* relations. The first three examples on the right are templates which all explicitly mention the *capital-of* relationship, but they offer more linguistic context than typical manually defined templates, which makes the sentences more natural. In general, we have found that BERT tends to struggle with shorter sentences. There are also patterns that give more implicit evidence of capital-of relationships, such as the two last ones for the *capital-of* relation. These capture indirect evidence, e.g. the fact that embassies are usually located in the capital of a country.

Given a set of templates that were selected after the filtering step, one can then use these templates to perform link prediction, which is the task of finding a tail word t, given some source word s, such that

Currency	Capital-of
Sales of all products and services traded online in * in 2012 counted 311.6 billion *	Summer olympics, which were in *, the capital of the home country, *
As is often the case in *, lottery ticket prices above the 80 * threshold are negotiable	The main international airport serves *, the capital of and most populous city in *
The Government of * donated 300 million * to finance the school's construction in 1975	It is located in *, the capital of *
On his return to *, he had made 18,000 * on an initial investment of 4,500	In 2006, he portrayed John Morton on a tour of * arranged by the US Embassy in *
The cost of vertebroplasty in * as of 2010 was 2,500 *	At the time, Jefferson was residing in *, while serving as American Minister to *

Table 6: Automatically-extracted templates filtered by BERT associated with the *currency* and *capital-of* relations from the Google analogy dataset.

(s, t) is an instance of a considered relation. To find plausible tail words, one can simply aggregate the predictions that are made by a masked LM for the sentences $\phi_1(s, \_), ..., \phi_k(s, \_)$. For relation induction, more specifically, given a candidate pair (s, t) the problem is to determine whether (s, t) is likely to be a correct instance of the considered relation. In this case, it is not sufficient that t is predicted for some sentence $\phi_i(s, \_)$. To illustrate this, consider the following non-sensical instantiation of a capital-of template:

The capital of Macintosh is ___.

One of the top predictions by the BERT-Large-Uncased model is *Apple*, which might lead us to conclude that (*Macintosh*, *Apple*) is an instance of the capital-of relation. Rather than trying to classify a given word pair (s, t) by filling in MASK, in [Bouraoui *et al.*, 2020b] we used the full sentence $\phi(s, t)$ as input to the BERT LM, and train a classifier on top of the output produced by BERT. In particular, the output vector for the [CLS] token is used for this purpose, which has been shown to capture the overall meaning of the sentence [Devlin *et al.*, 2019]. The vector $h_{[CLS]}$ predicted for the [CLS] token intuitively captures whether the input sentence is natural or unusual, and thus whether (s, t) is likely to be a valid instance of the relation. In particular, by adding a classification layer that takes the $h_{[CLS]}$ vector as input, we are able to predict whether the input sentence $\phi_i(s, t)$ is a correct assertion, i.e. whether the pair (s, t) is an instance of the considered relation.

Overall, the experimental analysis in [Bouraoui *et al.*, 2020b] show that high-quality relational knowledge can be obtained in a fully automated way, without requiring any hand-coded templates. However, the method is not suitable for all types of relations. In particular, as could be expected, we found that our method is not suitable for morphological relations. We also found that it performs similarly to the methods prescribed in Section 3.1 that rely on pre-trained word vectors when it comes to lexical relations such as meronymy and hypernymy. However, for relations that require encyclopedic or commonsense knowledge, we found that our model consistently, and often substantially, outperformed methods relying on word vectors. This shows that the BERT language model indeed captures commonsense and factual knowledge to a greater extent than word vectors, and that such knowledge can be extracted from these models in a fully automated way.

3.3 Unsupervised Learning of Distributional Relation Vectors

Most word embedding models represent words as vectors, such that similar words are represented as similar vectors (e.g. in terms of cosine similarity or Euclidean distance). As discussed in the previous section, a remarkable property of these models is their capability to capture various lexical relationships, beyond mere similarity. However, vector translation models, and word embeddings as a source of semantic knowledge contain some limitations in capturing relational knowledge. In word embedding, a vector $\mathbf{w}$ of word w represents it in terms of its most salient features. For example, $\mathbf{w}_{paris}$ would implicitly encodes that Paris is located in France and that it is a capital city, etc. The $\mathbf{w}_{france}$ will capture that France is a country and is located in Europe. Most of the salient features in which Paris and France differ are related to the fact that the former is a capital city and the latter is a country, which is intuitively why the 'capital of' relation can be modeled in terms of a vector translation. Other relationships, however, such as the fact that Macron succeeded Hollande as president of France, are unlikely to be captured by word embeddings.

In [Jameel *et al.*, 2018], we considered the problem of learning a relation vector r_{ik} that capture how a source word i is related to a target word k in an unsupervised way from corpus statistics. The model intuitively captures which context word j are most closely associated with the word pair (i, k). To this end, we proposed a variant of GloVe model in which word vectors can be directly interpreted as PMI-weighted bag-of-words representations. The GloVe model is based on statistics about (*main word*, *context word*) pairs. In our setting, we reled on statistics about (*source word*, *context word* and *target word*). To capture co-occurrence statistics among three words, we proposed a generalization of PMI to three arguments. As showed in the experiment conducted in [Jameel *et al.*, 2018], such vectors can be used to find word pairs that are similar to a given word pair (i.e. finding analogies), or to find the most prototypical examples among a given set of relation instances. They can also be used as an alternative to relation extraction methods, by subsequently training a classifier that uses the relation vectors as input, which might be particularly effective in cases where only limited amounts of training data are available (with the case of analogy finding from a single instance being an extreme example). Moreover, the learned relation vectors can be used in various ways to enrich the input to neural network models. As a simple example, in a question answering system, mentions of entities could be annotated with relation vectors encoding their relationship to the different words from the question. As another example, in recommendation system by taking advantage of vectors expressing the relationship between items that have been bought (or viewed) by a customer and other items from the catalogue. Finally, relation vectors should also be useful for knowledge completion, especially in cases where few training examples per relation type are given (meaning that neural network models could not be used) and where relations cannot be predicted from the already available knowledge (meaning that knowledge graph embedding methods could not be used, or are at least not sufficient).

3.4 Conclusion

This chapter covers three ways for modelling relational knowledge with (contextualiszd) word vector representations. First, based on the common assumption that lexical relations correspond to vector translations in a word embedding, probabilistic models can be used for identifying word pairs that are in a given relation. Second, we studied how high-quality relational knowledge can be obtained in a fully automated way from pre-trained language models such as BERT, without requiring any hand-coded templates. Finally, we proposed methods that use co-occurrence statistics to represent the relationship between a given pair of words as a vector.

CHAPTER 4

DERIVING WORD VECTORS FROM CONTEXTUALISED LANGUAGE MODELS

In the last few years, Word Embeddings have been largely used as a form of prior knowledge for many applications where word meaning has to be modelled in the absence of (sentence) context. For instance, in information retrieval, query terms often need to be modelled without any other context, and word vectors are commonly used for this purpose [Onal *et al.*, 2018]. In entity retrieval, using word vectors is particularly crucial to match query terms to the vector encodings of candidate entities [Nikolaev and Kotov, 2020]. In zero shot learning, word vectors are used to obtain category embeddings [Socher *et al.*, 2013]. Word vectors are also used in topic models [Das *et al.*, 2015]. In the context of the Semantic Web, word vectors have been used for ontology alignment [Kolyvakis *et al.*, 2018], concept invention [Vimercati *et al.*, 2019] and ontology completion [Li *et al.*, 2019]. The word vectors learned by standard word embedding models, such as Skip-gram and GloVe, essentially summarise the contexts in which each word occurs. However, these contexts are modelled in a shallow way, capturing only the number of co-occurrences between target words and individual context words. Recently, contextualized language models such as BERT [Devlin *et al.*, 2019] have largely replaced the use of static (i.e. non-contextualized) word vectors in many NLP tasks among others. The question we address in this chapter is whether the more sophisticated context encodings that are produced by LMs can be used to obtain higher-quality word vectors.

4.1 Modelling General Properties

As a simple strategy to obtain static word vectors from a contextualised language model, we can sample sentences in which a given word w occurs, obtain a contextualised vector representation of w from these sentences, and finally average these vectors as suggested in [Bommasani *et al.*, 2020; Vulic *et al.*, 2020].

Given W a set of words for which we want to learn a vector representation, we proposed in [Li *et al.*, 2021] a method that first consists in randomly sampling N mentions of each $w \in W$ from a given corpus. From each of the corresponding sentences, a vector representation is obtained by masking the occurrence of w and taking the contextualised vector predicted by BERT for the position of this [MASK] token. We refer to this obtained vector as a mention vector. In particular, this method has a least two advantages. First, given a word w, considering [MASK] allows us to specifically capture what

Masked sentence	BERT predictions
___ *are cultivated by both small farmers and large land holders.*	they, these, crops, fields, potatoes, gardens, vegetables, most, vines, trees
Banoffee pie is an English dessert pie made from ___, cream and toffee ...	cheese, sugar, butter, apples, eggs, milk, chocolate, honey, apple, egg
___ *are a popular fruit consumed worldwide with a yearly production of over ...*	they, bananas, citrus, apples, grapes, these, fruits, potatoes, berries, nuts

Table 4.1: Top predictions from BERT-large-uncased for sentences.

each sentence reveals about the word w. Second, since w is replaced by a single [MASK] token, a single vector is always obtained, even if w corresponds to multiple sub-word tokens. In contrast, without masking, the predictions for the different sub-word tokens from the same word have to be aggregated in some way. In fact, by using the MASK token, $\mathbf{w}_{\text{AVG}}$ reflects the properties of w that can be inferred from typical sentences mentioning w, rather than the properties that best discriminate w from other words. A bag of masked sentences can thus be viewed as a bag of properties. As result, the static vectors $\mathbf{w}_{\text{AVG}}$ are considered qualitatively different from the vectors that are obtained by standard word embedding models. This is illustrated in Table 4.1, showing Wikipedia sentences where occurrences of the word *bananas* were masked. From BERT's top predictions for the missing word, we can see that these sentences indeed reveal different properties of bananas, e.g. being edible, a dessert ingredient and a type of fruit.

4.2 Filtering and Sentence Selection Strategies

The view of masked sentences as encoding properties suggests another improvement over plain averaging of contextualised vectors. Since some properties are intuitively more important than others, the final representations could be then improved by averaging only a particular selection of the contextualised vectors. More precisely, the mention vectors are affected by factors that are irrelevant for our purposes, such as the syntactic role of the word in the sentence, or its position. Second in the aforementioned method, a large number of sentences were used for each word, which is computationally expensive for many settings, especially those with a large vocabulary. When masking w, the resulting vector representation can only capture what the sentence reveals about w. As most sentences are rather uninformative, we may thus wonder whether high-quality embeddings can be learned from just a few mentions of each word. A related research question is whether better results are possible by selecting sentences strategically rather than at random. In this section, we answer these questions by empirically analyzing a range of strategies for selecting mentions of a given word w. On the one hand, this is motivated by the practical desire to distil word vectors from language models in a more efficient way. On the other hand, comparing the ef-

Target	Masked sentence
banana	Some countries produce statistics distinguishing between ___ and plantain production, but four of ...
sardine	Traditional fisheries for anchovies and ___ also have operated in the Pacific, the Mediterranean, and ...
lamb	Edison's 1877 tinfoil recording of Mary Had a Little ___, not preserved, has been called the first ...
pineapple	In October 2000, the Big ___, a tourist attraction on the Sunshine Coast, was used as a backdrop for ...
salamander	The southern red-backed salamander (Plethodon serratus) is a species of ___ endemic to the United States.

Table 4.2: Examples of sentences whose corresponding mention vectors were filtered.

fectiveness of different sentence selection strategies can also provide us with insights into how language models acquire knowledge about word meaning.

Filtering idiosyncratic properties. To illustrate the intuition behind the sentence selection process, let us consider the following Wikipedia sentence: *Banana equivalent dose (BED) is an informal measurement of ionizing radiation exposure*. By masking the word "banana", one can obtain a contextualised vector, but this vector would not capture any of the properties that we would normally associate with bananas. The crucial difference with the sentences from Table 4.1 is that the latter capture *general properties*, i.e. properties that apply to more than one concept, whereas the sentence above captures an *idiosyncratic property*, i.e. a property that only applies to a particular word. Broadly speaking, we can distinguish between those that capture idiosyncratic properties (i.e. properties that only apply to a particular word) and those that capture more general properties. Inspired by this view, in [Li *et al.*, 2021], we proposed strategy for identifying contextualised vectors that are likely to capture idiosyncratic properties that we can simply remove when computing the average of the remaining ones. More precisely, for each mention vector $\mathbf{m} \in \mu(w)$, the idea is to compute its k nearest neighbours, in terms of cosine similarity, among the set of all mention vectors that were obtained for the vocabulary W, i.e. the set $\bigcup_{v \in W} \mu(v)$. If all these nearest neighbours belong to $\mu(w)$ then we assume that $\mathbf{m}$ is too idiosyncratic and should be removed. Indeed, this suggests that the corresponding sentence expresses a property that only applies to w. We then represent w as the average $\mathbf{w}^*$ of all remaining mention vectors, i.e. all mention vectors from $\mu(w)$ that were not found to be idiosyncratic. Table 4.2 provides some examples of sentences whose resulting mention vector was filtered, for words from X-McRae feature norms dataset McRae et al. [2005]. The sentence for *banana* asserts a highly idiosyncratic property, namely that the words *banana* and *plantain* are interchangeable in some contexts. The example for *sardine* is filtered because *sardines* and *anchovies* are often mentioned together. The examples for *lamb* and *pineapple* illustrate cases where the target word is used within the name of an entity, rather than on its own. Finally, as the example for *salamander* illustrates, highly idiosyncratic vectors can be obtained from sentences in which the target word is mentioned twice.

Selecting Topic-Specific Mentions In [Wang *et al.*, 2021] we suggested that random strategies may not be optimal. Intuitively, this is because the contexts in which a given word w is most frequently mentioned might not be the most informative ones, i.e. they may not be the contexts which best characterize the properties of w that matter for a given task. We suggested a strategy based on topic models, which first consists in identifying the topics which are most relevant for the target word w and then for each of the selected topics t, select sentences $s_1^t, ..., s_n^t$ mentioning w from documents that are closely related to this topic. For each of the selected topics t, the sentences $s_1^t, ..., s_n^t$ are then used to construct a topic-specific vector $\mathbf{w}^t$, using some strategies. The final representation of w will be computed as a weighted average of these topic-specific vectors. The topics can be obtained using Latent Dirichlet Allocation (LDA) [Blei *et al.*, 2003], which aims to obtain a representation of each document d in the considered corpus as a multinomial distribution over m topics. The weighted average of the different input vectors uses a task specific supervision signal. In particular, given $\mathbf{w}_1, ..., \mathbf{w}_k$ the different vector representations available for word w (e.g. the vectors from different transformer layers), the combination of these vectors is done by computing a weighted average learned with the model in which $\mathbf{w}$ is used.

Topic-specific vectors can then be expected to focus on different properties, depending on the chosen topic. Table 4.3 lists, for a sample of words from the WordNet supersenses dataset [Ciaramita and Johnson, 2003], the top 5 nearest neighbours per topic in terms of cosine similarity. We can see that for the word *'partner'*, its topic-specific embeddings correspond to its usage in the context of 'finance', 'stock market' and 'fiction'. These three embeddings roughly correspond to three different senses of the word[1]. This de-conflation or implicit disambiguation is also found for words such as *'cell'*, *'port'*, *'bulb'* or *'mail'*, which shows a striking relevance of the role of mail in the election topic, being semantically similar in the corresponding vector space to words such as 'telemarketing', 'spam' or 'wiretap'. In the case of *'fingerprint'*, we can also see some implicit disambiguation (distinguishing between fingerprinting in computer science, as a form of hashing, and the more traditional sense). However, we also see a more topical distinction, revealing differences between the role played by fingerprints in fictional works and forensic research. This tendency of capturing different contexts is more evidently shown in the last four examples. First, for *'sky'* and *'strength'*, the topic-wise embeddings do *not represent different senses of these words*, but rather indicate different types of usage (possibly related to cultural or commonsense properties). Specifically, we see that the same sense of *'sky'* is used in mythological, landscaping and geological contexts. Likewise, *'strength'* is clustered into different mentions, but while this word also preserves the same sense, it is clearly used in different contexts: physical, as a human feature, and in military contexts. Finally, *'noon'* and *'galaxy'* (which only occur in two topics), also show this topicality. In both cases, we have representations that reflect their physics and everyday usages, for the same senses of these words.

Additional Selection Strategies We now discuss a number of alternative sentence selection strategies we proposed in [Wang *et al.*, 2022] and aimed at providing us with more informative sentences to obtain

[1] In fact, we can directly pinpoint these vectors to the following WordNet [Miller, 1995] senses: `partner.n.03`, `collaborator.n.03` and `spouse.n.01`.

WORD	TOPIC	NEAREST NEIGHBOURS
partner	{research, professor, science, education, institute}	beneficiary, creditor, investor, employer, stockholder
	{football, republican, coach, senate, representatives}	lobbyist, bookkeeper, cashier, stockbroker, clerk
	{game, book, novel, story, reception}	nanny, spouse, lover, friend, secretary
cell	{protein, disease, medical, cancer, cells}	lymphocyte, macrophage, axon, astrocyte, organelle
	{food, plant, water, gas, power, oil}	electrode, electrolyte, cathode, anode, substrate
	{physics, mathematics, space, ngc, theory}	surface, torus, mesh, grid, cone
port	{station, building, railway, historic, church}	harbor, seaport, dock, waterfront, city
	{radio, station, fm, software, data, forewings}	link, gateway, router, line, socket
	{game, book, novel, story, reception}	version, remake, compilation, patch, modification
bulb	{station, building, railway, historic, church}	lamp, transformer, dynamo, projector, lighting
	{protein, disease, medical, cancer, cells}	epithelium, ganglion, nucleus, gland, cortex
	{species, genus, described, description, flowers}	rootstock, fern, vine, tuber, clover
mail	{station, building, railway, historic, church}	cargo, grain, baggage, coal, livestock
	{game, book, novel, story, reception}	paper, jewelry, telephone, telegraph, typewriter
	{party, election, minister, elected, elections}	telemarketing, spam, wiretap, internet, money
fingerprint	{radio, station, fm, software, data, forewings}	signature, checksum, bitmap, texture, text
	{game, book, novel, story, reception}	cadaver, skull, wiretap, body, tooth
	{party, election, minister, elected, elections}	wiretap, forensics, postmortem, polygraph, check
sky	{greek, ancient, castle, king, roman}	underworld, sun, afterlife, zodiac, moon
	{river, lake, mountain, island, village}	horizon, ocean, earth, sun, globe
	{physics, mathematics, space, ngc, theory}	ionosphere, sun, globe, earth, heliosphere
strength	{food, plant, water, gas, power}	stiffness, ductility, hardness, permeability, viscosity
	{game, book, novel, story, reception}	intelligence, agility, charisma, power, telepathy
	{army, regiment, navy, ship, air}	morale, firepower, resistance, force, garrison
noon	{physics, mathematics, space, ngc, theory}	declination, night, equinox, perihelion, latitude
	{army, regiment, navy, ship, air}	dawn, sunset, night, morning, shore
galaxy	{physics, mathematics, space, ngc, theory}	nebula, quasar, pulsar, nova, star
	{game, book, novel, story, reception}	globe, future, world, planet, nation

Table 4.3: Nearest neighbours of topic-specific embeddings for a sample of words from the WordNet SuperSenses dataset, using BERT-base embeddings. The top 6 selected samples illustrate clear topic distributions per word sense, and the bottom 4 also show topical properties within the same sense. The most relevant words for each topic are shown under the **TOPIC** column.

high-quality word vectors from a small number of sentences, which is essential for scaling up the methods for distilling word embeddings. Given this focus on efficiency, [Wang *et al.*, 2022] first considered two strategies that rely on the structure of Wikipedia:

- INTRO: The sentences are only sampled from the introductory section of a Wikipedia article (regardless of what the article is about). The intuition is that these introductory sections are more likely to contain sentences in which properties of words are mentioned explicitly.
- HOME: If there is a Wikipedia article about a word w, then select the first n sentences mentioning w from that article. If w does not have a Wikipedia, then fall back on RAND (Random selection).

In addition, [Wang *et al.*, 2022] proposed a number of strategies that rely on aspects of the sentences themselves:

- POS: only sample sentences which start with the word w in plural form. The intuition is that such sentences are likely to express generic knowledge about w.
- ENUM: first select all sentences in which w is preceded or succeeded by a comma or the word 'and'. Then rank these sentences based on the number of commas, as a simple strategy for prioritizing longer enumerations, and select the n highest ranked sentences. The intuition is that enumerations can provide us with useful knowledge, capturing the fact that the words in the enumeration have some property in common with w.
- PMI: For all words that co-occur with w in at least 2 sentences, compute their Pointwise Mutual Information (PMI) in an offline preprocessing step. This PMI score reflects to what extent these words appear more often in the same sentence than would be expected by chance, given their overall frequency. Given a target word w, we first identify the n words whose PMI score with w is highest. For each of these n words, we then randomly select one sentence mentioning that word.

Finally, beyond Wikipedia, [Wang *et al.*, 2022] consider two external sources for obtaining sentences, because of their focus on generic knowledge.

- DEF: extract the (primary) definition of w from the English fragment of Wiktionary[2].
- GENERIC: first select all sentences about w in GenericsKB [Bhakthavatsalam *et al.*, 2020] that originate from a text corpus[3]. The sentences are then ranked based on their confidence score in GenericsKB and select the top n.

For all strategies, if there are fewer than n sentences that can be selected, then we fall back to RAND for the remaining sentences. Based on our analysis provided in [Wang *et al.*, 2022], the most effective strategies are to select sentences using PMI and to include a definition of the target word. The success

[2] `https://www.wiktionary.org`

[3] GenericsKB also contains sentences that were generated from knowledge graph triples. Given the short and artificial nature of these sentences, there are not considered.

of these strategies makes it possible to use word embeddings obtained from LMs in applications such as ontology completion and zero shot learning with minimal computational overhead. Section 4.3 presents our work on few-shot image classification. Ontology completion is described in the next chapter.

4.3 Application to Few-Shot Learning

Multi-label image classification (ML-IC) has received considerable attention in recent years [Wang *et al.*, 2016; Chen *et al.*, 2019a; Wang *et al.*, 2017; Yazici *et al.*, 2020]. This task aims to assign descriptive labels to images, where each image is typically associated with multiple labels. Standard approaches for this task often focus on modelling label dependencies, e.g. taking advantage of the fact that the presence of one label makes the presence of another label more (or less) likely. In the few-shot setting, however, we only have a small number of images available for training, possibly only a single image for some labels. In this setting, only relying on label co-occurrence statistics is not feasible. The problem of few-shot image classification (FSIC), i.e. image classification with limited training data in the single-label setting, has also received considerable attention. However, standard approaches for this task are not suitable for the multi-label setting. For instance, so-called metric-based approaches learn a prototype for each image category, and then assign images to the category whose prototype is closest to the image in some sense. These prototypes are typically obtained by averaging a representation of the training images. In the seminal ProtoNet model [Snell *et al.*, 2017], for instance, prototypes are simply defined as the average of the global feature maps of the available training examples. This strategy crucially relies on the assumption that most of the image is somehow relevant to its category. In the multi-label setting, however, such an assumption is highly questionable, given that different labels tend to refer to different parts of the image. For instance, given an image depicting a car and a bike, using a representation of the entire image to obtain a prototype for bike would be misleading. When estimating prototypes, in a metric-based setting, it is thus important to determine which regions are relevant for which labels, but the limited amount of training data makes this highly challenging. As a solution, in [Yan *et al.*, 2021a,b, 2022] we proposed to use static word embeddings learned using aforementioned (Section 4.1 and 4.2) as a form of prior knowledge about the meaning of the labels.

Aligning Visual Prototypes with BERT Embeddings Two notable examples of models that rely on class names are AM3 [Xing *et al.*, 2019] and TRAML [Li *et al.*, 2020], both of which use the GloVe [Pennington *et al.*, 2014] word embedding model for representing class names. However, standard word vectors, such as those from GloVe, are strongly influenced by topical similarity. This is illustrated in Table 4.5, which shows the top-3 most similar classes from miniImageNet for three example targets. For instance, the nearest neighbours of *catamaran* include *snorkel* and *jellyfish*. These words are all clearly topically related, but catamarans are not *similar* to *snorkels* or *jellyfish*. This is problematic for few-shot learning, where we would intuitively want that class names with similar embeddings denote categories of the same kind. To address this issue, we use vectors learned from Section 4.1. We qualitatively observe that the resulting embeddings are indeed better suited for grouping classes that are conceptually

	banana
RAND	• Born in Puntarenas Province , Lagos ' parent decided to move to Limón where Cristhian went to school and worked in banana plantation • Binding post or banana plug may be used for lower frequency • In India , vegetarian variety may use potato , calabash , paneer , or banana •A later claim suggested that Bubbles had died ; Jackson's press agent Lee Solters quipped to the medium that when Bubbles heard about his demise he went banana ... Like Mark Twain , his death is grossly exaggerated and he 's alive and doing well • At the Royal Variety Performance in 1981 , it was performed in the customary male evening dress by Anita Harris , who brought the house down with the line "I've just had a banana with Lady Diana" in the Buckingham Palace verse of the song
HOME	•A banana is an elongated, edible fruit – botanically a berry – produced by several kinds of large herbaceous flowering plants in the genus "Musa" • In some countries, bananas used for cooking may be called "plantains", distinguishing them from dessert bananas • Almost all modern edible seedless (parthenocarp) bananas come from two wild species – "Musa acuminata" and "Musa balbisiana" • The scientific names of most cultivated bananas are "Musa acuminata", "Musa balbisiana", and "Musa" × "paradisiaca" for the hybrid "Musa acuminata" × "M. balbisiana", depending on their genomic constitution. • They are grown in 135 countries, primarily for their fruit, and to a lesser extent to make fiber, banana wine, and banana beer and as ornamental plants
INTRO	• The area produces citrus, olives, tomatoes and market-garden vegetables, and is one of the few parts of Europe where commercial banana production is possible. • The work, created in an edition of three, consists of a fresh banana taped to a wall with a piece of duct tape • They also sell orange, grape, piña colada, coconut champagne (non-alcoholic), and banana daiquiri (non-alcoholic) fruit drinks • No banana plantation was left unscathed by the hours-long onslaught of strong winds • The crops of highest productivity are plantain, banana, coconut, tomatoes, pepper, eggplant, yucca, rice, beans, maize, "guandules" and sweet potato
PMI	• The common fruits that are used in the preparation include banana, apple, kiwi, strawberry, papaya, pineapple, mango, and soursop • Thus the banana producer and distributor Chiquita produces publicity material for the American market which says that "a plantain is not a banana" • One day Mitchell posted a photo of herself on Twitter next to a bruised banana in response to trolls who had compared her freckles to the overripe fruit • The most important Philippine cooking banana is the saba banana (as well as the very similar cardava banana) • Their meals consist of cooked or steamed rice wrapped in banana or tara or kau leaves that known as "khau how" and boiled vegetables
POS	• Bananas, grown mainly for domestic consumption, amount to a steady annual average crop of 70,000 tons. • Bananas were introduced into the americas in the 16th century by portuguese sailors who came across the fruits in west africa, while engaged in commercial ventures and the slave trade" • Bananas must be transported over long distances from the tropics to world markets • Bananas was edited at the time by the now-legendary horror author r. l. stine • Bananas which are turning yellow emit natural ethylene which is characterized by the emission of sweet scented esters
ENUM	• Crops are, for example, cereals (mainly wheat, barley, rye and triticale), soybeans, banana, rice, coffee, turnips, and red as well as sugar beets • These have included: bacon maple ale and chocolate, peanut butter, and banana ale • There are also wild relatives of jackfruit, mango, cardamom, turmeric and banana • Amelita's signature dish was an organic rib fillet with shaved ham, banana, and hollandaise sauce. • Whereas the larger farming plots are utilized for staple crops, families can choose to grow herbs, flowers and fruit trees (mango, banana, plum, orange, lime) in their personal household garden
GENERIC	• Bananas contain more digestible carbohydrates than any other fruit • Bananas have no fat, cholesterol or sodium • Bananas do contain serotonin • Bananas grow on plants • Bananas contain pectin, a soluble fibre
DEF	• Banana is an elongated curved tropical fruit that grows in bunches and has a creamy flesh and a smooth skin

Table 4.4: Example sentences selected for the word *banana*.

	catamaran	**house finch**	**horizontal bar**
GloVe	snorkel yawl jellyfish	ladybug komondor triceratops	pencil box aircraft carrier beer bottle
BERT	yawl aircraft carrier school bus	goose toucan ladybug	parallel bars unicycle ear
$\text{BERT}_{\text{proj}}$	yawl school bus aircraft carrier	toucan robin ladybug	parallel bars scoreboard street sign

Table 4.5: Most similar miniImageNet classes to *house finch*, *horizontal bar* and *catamaran*, according to class name embeddings obtained using GloVe, BERT and the proposed projection of the BERT embeddings onto a 50-dimensional space ($\text{BERT}_{\text{proj}}$).

similar. For instance, as can be seen in Table 4.5, with the proposed BERT embeddings, the top 2 nearest neighbours are now also boats (being the only remaining boat classes in miniImageNet), while the third neighbour is also a vehicle. Furthermore, as the example of *house finch* shows, the BERT embeddings also tend to model semantic relatedness at a finer-grained level: while the top neighbours for GloVe are all animals, none of them are birds. In contrast, the top two neighbours for BERT are birds.

One disadvantage of BERT embeddings is that they are high dimensional, a problem which is exacerbated when using concatenations of several types of class name embeddings. Furthermore, we can expect that only some of the information captured by the class name embeddings may be relevant for image classification. In [Yan *et al.*, 2021a,b], rather than predicting visual prototypes from the class names, we model the visual and text-based prototypes separately. Moreover, we also proposed a dimensionality reduction strategy, inspired by work on aligning cross-lingual word embeddings [Artetxe *et al.*, 2018], which aims to find a subspace of the BERT embeddings that is maximally aligned with the visual prototypes. As illustrated in Table 4.5, the resulting embeddings remain at least as useful as the original BERT embeddings, despite only being 50-dimensional. In fact, some of the nearest neighbours for the low-dimensional vectors are arguably better than those of the BERT embeddings themselves, e.g. *toucan* is more similar to *house finch* than *goose* is, while *scoreboard* and *street sign* are more meaningful neighbours of *horizontal bar* than *unicycle* and *ear*. More precisely, for a given episode, the model we proposed first use the labelled images to construct visual prototypes, as in existing approaches. Each of the class names is represented by a vector that was learned from some text corpus using strategies devlopped in Sections 4.1 and 4.2. Both the visual prototypes and the class name embeddings feed into the Correlation Exploration Module (CEM), whose aim is to find a low-dimensional subspace of the class name embeddings. The resulting textual prototype is then used in combination with the visual prototype for making the final prediction.

Inferring Prototypes with Word Vector Guided Attention In multi-label image classification setting, we only used word vectors to identify which regions of the training images are most likely to be relevant

for a given label [Yan *et al.*, 2022]. As an example to explain the intuition of how word vectors can be useful for this purpose, assume that we have a number of labels that refer to animals. These labels will have similar word vectors, which tells the model that the predictive visual features for these different labels are likely to be similar. Now suppose we have an image which is labelled with *cat*. Based on training data for other labels, the model will select areas that are likely to contain an animal (although it would not necessarily be able to distinguish between cats closely related animals). Again, the word embeddings are used here as prior knowledge about the similarity between different labels, rather than for predicting visual prototypes. An important practical advantage of our method is that we can apply the model to previously unseen labels, without the need for any fine-tuning of the model's parameters on the novel label set. The model we devlopped in [Yan *et al.*, 2022] consists of two main components. The first component is aimed at jointly representing label embeddings and visual features in the same vector space. Essentially, this component aims to predict visual prototypes from the label embeddings. Since such prototypes are noisy, we do not use them directly for making the final label predictions. This component is merely used to learn a joint representation of visual features and labels. The second component is aimed at computing the final prototypes, by aggregating the local features of the corresponding support images based on an attention mechanism, which relies on the label representations that are obtained by the first component. Finally, to classify a query image, we project it to the joint embedding space and then compare it with the learned prototypes.

4.4 Conclusion

The problem of learning word vectors has already received considerable attention. However, previous work has mostly focused on the performance of such vectors in NLP tasks, where static word vectors have now largely been superseded by the use of pre-trained neural language models. In many other applications (e.g. few-shot learning), however, word vectors remain important, because they capture prior knowledge about the commonalities between different entities (e.g. category labels, query terms, predicate names). It is currently less well-understood how word vectors for such applications can best be learned. We have analysed the potential of averaging the contextualised vectors predicted by BERT to obtain high-quality static word vectors. When the MASK encoder is used, the resulting vectors tend to represent words in terms of the general semantic properties they satisfy, which is useful in tasks where we have to identify words that are of the same kind, rather than merely related. We have also proposed a filtering strategy to obtain vectors that de-emphasise the idiosyncratic properties of words, leading to improved performance in the considered tasks. Using a large number of sentences for each word to compute its mention vectors is computationally expensive. Given this focus on efficiency, we proposed several strategies for selecting sentences. Finally, we studied how these static vectors distilled from LMs can be used in the context of few shot learning.

CHAPTER 5

PLAUSIBLE REASONING ABOUT ONTOLOGIES

Commonsense knowledge is playing an increasingly important role in the development of AI systems. Such knowledge is available, for example, in large open-domain terminological knowledge bases such as Cyc or SUMO as ontological knowledge, in knowledge graphs (KGs) such as DBpedia and WikiData, as semantic markup (e.g. RDFa). Ontologies play an important role in areas such as Semantic Web [Homburg *et al.*, 2020], Information Retrieval [Chen *et al.*, 2019b], Natural Language Processing [Rospocher and Corcoglioniti, 2018], and machine learning [Hohenecker and Lukasiewicz, 2020], among others. However, the available ontologies (and KGs, as simple ontologies) are inevitably incomplete, where several rules and facts are missing. Several methods have been proposed for automated ontology (KG) completion [Beltagy *et al.*, 2013; Rocktäschel and Riedel, 2017] that exploit statistical regularities in a given ontology to predict plausible missing rules or facts. Unfortunately, meaningful knowledge is difficult to predict, especially since we have few examples of facts or rules. However, as most of the existing approaches are mainly based on inductive approaches, the resulting predictions might be conflicting with others. Section 5.1 provides an overview on methods for automatically finding missing ontological rules. In the same perspective, to widen the coverage of terminological knowledge to several domains and to deal with incompleteness, one may combine knowledge from several sources. However, it turns out that merging open-domain knowledge bases is a particularly challenging task as pointed out, for example, in [Tanon *et al.*, 2016] reporting the different problems and difficulties encountered when merging Freebase with WikiData. Section 5.2 provides with methods for ontologies merging that aims to combine two (or more) ontologies having the same terminology while handling conflict, using the conceptual spaces view. Conflicting information may occur when the statements of several sources are simply gathered together. To deal with conflicting knowledge, one can use a repair-based mechanism to maintain the consistency (A set of terminological statements (axioms)) is (logically) consistent, iff all the statements can be true together or it involves no contradiction., i.e., ensure that there are no conflicting (or contradictory) statements. Section 5.3 describes some methods for repairing ontologies, with a focus on inconsistency-tolerant query answering.

5.1 Automated Ontology Completion

The main underlying idea behind the approach we proposed in [Bouraoui and Schockaert, 2019], is that ontologies often contain large sets of rules which only differ in one predicate. As a simple example, consider the following rules

$$\textit{Beer}(x) \rightarrow R(x)$$
$$\textit{Gin}(x) \rightarrow R(x)$$

Without knowing what the predicate R represents, we can infer that the following rule is also valid:

$$\textit{Wine}(x) \rightarrow R(x)$$

This is intuitively because almost all natural properties which beer and gin have in common are also satisfied by wine. These natural properties can be captured by vectors obtained using work mentioned in Chapter 4. Based on analysis in [Li *et al.*, 2021; Wang *et al.*, 2022], static word vectors distilled from LMs led to better results.

Template-based approach. To formalize this intuition, in [Bouraoui and Schockaert, 2019] we considered the notion of rule templates. A rule template ρ is a second-order predicate, which corresponds to a rule in which one predicate occurrence has been replaced by a placeholder. For instance, in the above example, we can consider a template ρ such that $\rho(P)$ holds if the rule $P(x) \rightarrow R(x)$ is valid, meaning that we would expect this rule to be entailed by the ontology if the ontology were complete. Given such a template ρ, we can consider the set of all instances $P_1, ..., P_n$ such that the corresponding rules $\rho(P_1), ..., \rho(P_n)$ are entailed by the given ontology. The main strategy for finding plausible rules proposed in [Bouraoui and Schockaert, 2019] then essentially consists in finding predicates P which are similar to $P_1, ..., P_n$. More precisely, the predicates are represented as vectors and it is assumed that each template ρ can be modelled as a Gaussian distribution over the considered vector space, i.e. the probability that $\rho(P)$ is a valid rule is considered to be proportional to $\mathcal{G}_\rho(\mathbf{p})$, with $\mathbf{p}$ the vector representation of P and $\mathcal{G}_\rho$ the Gaussian distribution modelling ρ. In addition to the templates described above, which are called *unary templates*, [Bouraoui and Schockaert, 2019] also considered *binary templates*, which correspond to rules in which two predicate occurrences have been replaced by a placeholder. While unary templates enable a strategy known as interpolation, using binary templates leads to a form of analogical reasoning, both of which are well-established commonsense reasoning principles.

Taking account of dependencies. Critical aspect of this strategy for ontology completion is how the vector representation of the predicates is obtained. The approach from [Bouraoui and Schockaert, 2019] relies on the combination of two types of vectors : (i) the word vector of the predicate name, obtained from a pre-trained static word embedding (a standard word embedding such as Glove or SG or derived from a language model such as BERT as described in the previous chapter); (ii) a vector representation

which is learned from the ontology itself, using a variant of the AnalogySpace method [Speer *et al.*, 2008]. However, it is not clear why the predicates that satisfy a given template should follow a Gaussian distribution in the considered vector space. Moreover, the way in which the predicate representations are constructed does not maximally take advantage of the available information. In particular, the approach based on the AnalogySpace method only relies on the known instances of the unary templates, i.e. binary templates are completely ignored for constructing the vector representations of the predicates. This is clearly sub-optimal, as knowing that $\rho(P, R)$ is a valid rule, for a given binary template ρ, intuitively tells us something about the semantic relationship between the predicates P and R, which should in turn allow us to improve our representation of P and R.

To this end, in [Li *et al.*, 2019] we introduced a new method for predicting plausible rules based on Graph Convolutional Networks (GCNs). The main aim of GCNs is to represent and capture relationships among data. The proposed method starts from a graph-based representation of the rule base, in which the nodes correspond to predicates. Each node is annotated with a vector representation of the corresponding predicate. Crucially, however, rather than using these vectors directly for making predictions as in [Bouraoui and Schockaert, 2019], they are merely used for initializing the GCN. Edges are annotated with the binary templates that are satisfied by the corresponding pair of predicates. The proposed GCN model, which iteratively refines the vector encoding of the nodes, taking advantage of the edge annotations based on the binary templates. The resulting node vectors are then used to predict which predicates satisfy the different unary templates and which pairs of predicates satisfy the different binary templates, and thus to predict which rules are plausible. Note in particular, that the main aim of the GCN is to *learn* a vector representation of the predicates which is predictive of plausible rules, rather than relying on assumptions about a given vector representation. In the following the performance of the GCN model for ontology completion is illustrated using some examples of predicted rules from existing ontologies such as Wine and SUMO. As an example from the unary template setting, the model from [Li *et al.*, 2019] was able to correctly predict the following rule from the Wine ontology:

$$\textit{DryRedWine}(x) \rightarrow \textit{TableWine}(x)$$

by using the template $\rho(\star) = \star(x) \rightarrow \textit{TableWine}(x)$. The instances of this template that were given in the training data are *RedTableWine*, *DryWhiteWine* and *Burgundy*. Based on these instances, the Bayesian model from [Bouraoui and Schockaert, 2019] was not able to predict that *DryRedWine* is also a plausible instance. The GCN models, however, were able to exploit edges (i.e. binary templates) corresponding to

the following rules:

$$\begin{aligned} \textit{Merlot}(x) &\rightarrow \textit{DryRedWine}(x) \\ \textit{Merlot}(x) &\rightarrow \textit{RedTableWine}(x) \\ \textit{DryRedWine}(x) &\rightarrow \textit{DryWine}(x) \\ \textit{DryWhiteWine}(x) &\rightarrow \textit{DryWine}(x) \\ \textit{Burgundy}(x) &\rightarrow \textit{DryWine}(x) \end{aligned}$$

As an example from the binary template setting, the GCN model was able to correctly predict the following rule from the Olympics ontology:

$$\textit{WomansTeam}(x) \rightarrow \exists y \,.\, \textit{hasMember}(x, y) \wedge \textit{Woman}(y)$$

based on the following rules from the training data:

$$\begin{aligned} \textit{MensTeam}(x) &\rightarrow \exists y \,.\, \textit{hasMember}(x, y) \wedge \textit{Man}(y) \\ \textit{MixedTeam}(x) &\rightarrow \exists y \,.\, \textit{hasMember}(x, y) \wedge \textit{Woman}(y) \end{aligned}$$

This illustrates the ability of models based on binary templates to perform analogical reasoning. Note that this rule cannot be predicted in the setting where only unary templates are used.

From a practical perspective, an important question is whether the GCN model is able to find rules which are missing from the existing ontologies, rather than merely identifying held-out rules. In the following, we present some examples of rules that were predicted by our model, but which cannot be deduced from the full ontologies. These predictions are based on a GCN model that was trained on the full ontologies. Some of the rules we obtained are as follows:

$$\begin{aligned} \textit{Cycle}(x) &\rightarrow \textit{LandVehicle}(x) \\ \textit{AgriculturalProduct}(x) &\rightarrow \textit{Product}(x) \wedge \textit{Exporting}(x) \\ \textit{CargoShip}(x) &\rightarrow \textit{Ship}(x) \wedge \textit{DryBulkCargo}(x) \end{aligned}$$

As can be seen, these rules intuitively make sense, which suggests that our approach could indeed be useful to suggest missing rules in a given ontology. Since there exists rule $\textit{Bicycle}(x) \rightarrow \textit{Cycle}(x)$ in the Transport ontology, which makes $\textit{Cycle}(x) \rightarrow \textit{LandVehicle}(x)$ plausible. $\textit{AgriculturalProduct}(x) \rightarrow \textit{Product}(x) \wedge \textit{Exporting}(x)$ is plausible, here "Exporting", according to the Economy ontology, is employed in international trade, because of the rules $\textit{Exporting}(x) \rightarrow \textit{ChangeOfPossession}(x)$ and $\textit{Exporting}(x) \rightarrow \textit{FinancialTransaction}(x)$.

5.2 Region-Based Merging of Open-Domain Ontologies

Another way to expand ontological knowledge is to gather knowledge from different sources. Let us consider an example to illustrate the merging problem. Assume that a first source says that the concept $Paper$ is disjoint with the concept $Document$, while another source says that every $Paper$ is a $Document$. Obviously enough, these two statements are conflicting. To be faithful to both sources while resolving conflicts, a sensible choice would be to assume that $Paper$ and $Document$ are not disjoint concepts, but every $Paper$ is not necessarily a $Document$, that is, the two concepts *partially overlap*. This kind of result is clearly consistent and can be seen as a good compromise between both sources. Finding a meaningful and relevant compromise between sources during the merging process is difficult. This is mainly due the fact that ontology languages (Description Logics for example) are not expressive enough to capture salient knowledge that might be needed during the merging process (See chapter 2). The simple example pointed out above clearly shows some pieces of knowledge that should be taken into account during the merging process, but that cannot be captured in the ontology language. The problem of ontology (or DL) merging is close to the problem of belief merging in a propositional setting [Benferhat *et al.*, 2019; Kumar and Harding, 2016; Wang *et al.*, 2012]. For instance, in [Benferhat *et al.*, 2019] we studied merging assertional bases in the *DL-Lite* fragment. They have determined the minimal subsets of assertions to resolve conflicts based on the inconsistency minimization principle. In [Bouraoui *et al.*, 2020c]) we proposed a model-based merging operator for merging EL ontologies which solves semantic conflicts that arise during the merging process. However, all the existing approaches rely on the formal encoding frameworks of the ontologies, which is not flexible enough to capture relevant knowledge that might emerge during the merging process.

Taking inspiration from conceptual spaces, in [Bouraoui *et al.*, 2022b] we introduced a novel method for merging open-domain terminological knowledge that relies Region Connection Calculus (RCC5), a formalism used to represent regions in a topological space and to reason about their set-theoretic relationships. Motivated by the fact that conceptual knowledge in an ontology can be to some extent modelled as geometric objects and constraints on metric spaces as shown in Chapter 2, the proposed method for ontology merging that takes advantage of *qualitative spatial reasoning* to find out a relevant compromise between sources while resolving conflicts. Qualitative spatial reasoning is a suitable paradigm for efficiently reasoning about spatial entities and their relationships, where knowledge is represented as a so-called *qualitative constraint network (QCN)*. Spatial information is usually represented in terms of basic or non-basic relations in a qualitative calculus, where reasoning tasks are then formulated as solving a set of qualitative constraints. In particular, the Region Connection Calculus (RCC) is a well-studied formalism for qualitative topological representation and reasoning, including its subsets *RCC-5* and *RCC-8* [Schockaert and Li, 2013]. Two significant advantages of the RCC framework are its ability to reason efficiently about the relationships between spatial entities, and its ability to deal with conflicts in qualitative constraint merging. Intuitively, the representation of region constraints into QCNs allows for more expressivity than when using DL rules (or constraints). In particular, QCNs are expressive enough to allows for disjunctions in the constraints. Several QCN merging operators have been intro-

duced in the literature. Roughly speaking, these operators compute a distance between QCN scenarios and the input QCNs. Then the scenarios with a minimal distance are selected as the best candidates for the merged result. Taking inspiration from these works, we proposed in [Bouraoui *et al.*, 2022b] to use RCC-5 formalism for merging open-domain terminological knowledge (simply called ontologies) using QCNs. They first show how to translate such knowledge into qualitative spaces while preserving its semantics and properties and then propose a merging operator that produces a single and consistent region space representing a compromise between sources. Finally, we shown how to express the region space in the input ontology language while maintaining all relevant information.

5.3 Conflict-Based Inconsistency Handling

Ontology-mediated query answering (OMQA) provides query reformulation techniques over ontological domain knowledge to improve access to data. While in OMQA, the ontological knowledge is assumed to be satisfiable, fully reliable, and often debugged by experts, the data (i.e., the assertional base) is usually of low quality. This for example may happen when collecting data from several sources [Bouraoui *et al.*, 2020c], or due to ontology evolution [Mohamed *et al.*, 2022a]. When the data is conflicting with the ontology, logical deduction performed for query answering is no longer appropriate, i.e, every fact can be derived as an answer to a query including the conflicting facts causing the inconsistency (ex falso quodlibet sequitur). Logical deduction is the key inference mechanism to draw sound conclusions from a knowledge base (ontology). It provides natural explanations for the consequences that can be derived. However, when the knowledge base is inconsistent, logical deduction is no longer appropriate, because it lead to a trivialization problem where every formula can be derived from an inconsistent base, including the conflicting information causing the inconsistency. The problem of inconsistency management has received considerable attention in a wide variety of areas, including databases (e.g [Bertossi, 2011]), multi-agent systems (e.g [Hunter *et al.*, 2014]), modal logics (e.g [Bouraoui *et al.*, 2020d]), belief merging and revision (e.g [Bouraoui *et al.*, 2022b]) where many approaches have been proposed to reason under inconsistency either by weakening the input base or weakening the deduction relation. To handling inconsistency in OMQA, several inconsistency-tolerant inference relations, called semantics, have been proposed. Most of these semantics, inspired by database reparation or nonmonotonic reasoning in propositional logic, consist in getting rid of inconsistency by first computing a set of (maximally) consistent subsets of the assertional base, called repairs, and then using them to perform query answering.

In different situations, information is often affected with uncertainty and imprecision. This is due for instance to the presence of a preference ranking between them reflecting their level of certainty, or the reliability of the sources that provides them. Representing such information generally gives rise to a prioritized (i.e. stratified) knowledge base. It is argued that handling priority or uncertainty is in complete agreement with possibility theory, which offers a very natural framework to deal with ordinal, qualitative uncertainty, preferences and priorities. It is particularly appropriate when the uncertainty scale only reflects a priority relation between different pieces of information. This is often the case in applications where not enough data is available to estimate a meaningful probabilistic representation. For instance,

[Mohamed *et al.*, 2018] we proposed an extension of EL ontologies using possibility theory. In some scenarios, the data are coming from different conflicting sources having different reliability levels. To take into account this information while handling inconsistency, several inconsistency-tolerant semantics have been considered based on the notion of preferred repairs when the assertions base is prioritized [Mohamed *et al.*, 2022c]. Inspired by word embedding models, in [Mohamed *et al.*, 2022c] another direction for handling inconsistency have been proposed based on exploiting co-occurrence conflicting relations for computing preferred repairs. When repairing conflicting ontologies, one important piece of information that needs to be considered is the participation of each assertion in the conflict and to what extent a fact is likely to be incompatible with each other. To provide an answer to this question, [Mohamed *et al.*, 2022c] exploit conflict regularities between facts by computing an embedding (a vector space representation) of the assertional base in which the distance between two facts reflects their compatibility. To represent the conflict between each pair of assertions, we use a one-hot matrix encoding of conflict, called conflict matrix. The conflict matrix takes the set of assertions as rows and columns. We assign 1 when there exists a conflict between the two assertions and 0 otherwise. When this matrix is computed, one can then apply several embedding techniques to represent data in a flexible way. For instance, one can apply the multidimensional Scaling [Cox and Cox, 2008] to obtain an embadding of the assertions where the distance between each pair of assertions represents their similarity. The embedding will serve then as a basis for computing the repairs. For instance, we can start with the assumption that each point (or assertion) is independent of the others and forms an individual cluster in the space. We then seek possible compatibilities between assertions, i.e., compatible clusters. If two assertions are similar, i.e., close in space and consistent with each other, then they are merged in the same cluster. After that, we obtain a set of clusters, each of which contains a set of consistent assertions.

5.4 Conclusion

We have proposed a method for inductive reasoning about description logic concepts, based on a vector space embedding of individuals (and concept names). We have also discussed how conceptual spaces can be used to solve merging or inconsistency problem in classical knowledge bases.

CHAPTER 6

PERSPECTIVES AND FUTURE RESEARCH DIRECTIONS

This chapter concludes the habilitation by summarising our contributions and presenting some future research directions.

Summary. We presented several contributions in different areas situated at the intersection between knowledge representation and reasoning and natural language processing. The central aim consisted in developing and implementing methods for reasoning based on symbolic and numerical representations. This includes works on learning and reasoning with conceptual space representations, which offer an interface between vector space embeddings and symbolic knowledge. Considerable attention has been recently devoted to the use of relational knowledge of the form (h, r, t) as background knowledge for developing AI systems. A part of our work was dedicated to modelling and inducing relational knowledge from vector space representations. With the emergence of contextulized language models such as BERT, GPT3 or RoBERTa, which have led to paradigm-shift in natural language processing, it was shown that these LMs capture many aspects of commonsense knowledge. To this end, we proposed several works on distilling commonsense knowledge from contextualised language models with the aim to produce high-quality vectors that can serve as prior knowledge about entities and labels for several applications such as few-shot learning and automated ontology completion. Finally, we have shown how both symbolic and sub-symbolic learning can be combined in a principled way for reasoning with ontologies.

Perspectives. In the following, we consider some perspectives for future research.

Conceptual spaces. Vector representations have many advantages compared to symbolic knowledge representation frameworks as they allow to model similarity and can be easily integrated with neural network models. But they also have important limitations when modelling concepts, properties and relations. While we proposed some approaches to identify plausible missing generic knowledge in ontologies, we need to develop principled mechanisms that tightly integrate induction with deductive reasoning. One possible answer is to extend conceptual spaces, modelling relations as regions in high-dimensional spaces and viewing rules as qualitative spatial constraints between these regions. Another possibility is to abstract away from actual conceptual space representations and develop a calculus for reasoning

about incomplete qualitative constraints on conceptual space representations (e.g. rules and betweenness assertions).

Rule Induction While vector space embeddings are clearly useful for plausible reasoning with ontologies, they also have a number of limitations, which we hope to address in future work. First, while entity embeddings offer a natural framework for modelling concepts, roles can only be modelled in a restricted way. For example, we typically do not have a vector space representation of a role name. Moreover, while the use of vector differences often leads to a meaningful representation of role instances, this will not be the case for all roles. Second, while logical connectives such as intersection and union have a natural counterpart in the vector space, this is not the case for negation. Indeed, prototype theory deals with natural categories, and the complement of such a natural category is typically not a natural category itself. Similarly, role restriction axioms do not have a real counterpart in the vector space. Finally, while we rely on a pre-trained entity embedding, it would be of interest to exploit the given description logic axioms (along with e.g. annotations provided with the ontology) when learning this embedding.

Learning knowledge from LMs Distilling high-quality vectors from language models have several benefits in particular for few-shot learning applications, where word vectors can provide with prior knowledge about the between entities and labels. However, the mention vectors obtained from a LM such as BERT have two key limitations, when it comes to modelling the semantic properties of concepts. First, the geometry of the BERT mention vectors has some counterintuitive properties. For instance [Cai *et al.*, 2020] found that mention vectors appear in clusters, and within each of these clusters the geometry is more "natural". Other authors have found that there are some directions that can skew the results of similarity computations [Timkey and van Schijndel, 2021]. Even aside from these considerations, as shown in Sections 4.1 and 4.2, the mention vectors can be easily affected by their syntax, e.g. whether the word appears as the noun or object of the sentence. To this end, it thus makes sense to "normalize" the mention vectors before inducing static word vectors from them. The challenge is that how to do this normalisation without any explicit supervision signal. One possible solution is to assume that two mention vectors $\mathbf{m_w}$ and $\hat{\mathbf{m}}_\mathbf{w}$, corresponding to the same target word w, are likely to express the same property if there is significant overlap between the top-neighbours of $\mathbf{m_w}$ and the neighbours of $\hat{\mathbf{m}}_\mathbf{w}$. Let $X = \{(\mathbf{m}^\mathbf{1}_{\mathbf{w_1}}, \hat{\mathbf{m}}^\mathbf{1}_{\mathbf{w_1}}), ..., (\mathbf{m}^\mathbf{k}_{\mathbf{w_k}}, \hat{\mathbf{m}}^\mathbf{k}_{\mathbf{w_k}})\}$ be the set of all pairs of mention vectors whose top-neighbours are sufficiently similar, where W contains such pairs for all words in the vocabulary (but the two mention vectors within each pair are always about the same word).

Learning logical knowledge from LMs The plausible reasoning methods considered that we developed can be applied to existing ontologies, but to widen their scope, it would be interesting to see how they can help in various ways to learn better logical theories from data learned from language models. Indeed, one of the main impediments to a more widespread use of symbolic knowledge is the lack of comprehensive ontologies/theories for many domains. While there has been considerable work on learning taxonomies/ontologies from text collections, we believe such approaches could be substantially

improved by having access to a robust model for plausible reasoning coupled with language models. A simple idea is to learn a rule based classifier (or a decision tree, where we can think of the branches as rules) or to apply ILP methods. Then apply the "rule extrapolation" method to find missing rules. This should allow us to come up with rules that have low support in the training data (meaning that normal methods would not include it) but that seem nonetheless plausible. A related idea is to learn rules for classes for which we don't have any training data, by interpolating/extrapolating rules for classes where we do have training data. This is essentially the problem of "zero shot learning" but the method would work quite differently from existing methods. From a vectors that we can learn from LM capturing properties we can learn default rules like "most films are entertaining". In this way, a comprehensive set of default (and hard) rules for a given domain could be obtained. Moreover, we can use these default rules as training data for a method that learns how to extract default rules from text documents. Learning to identify rules stated in text documents is a promising approach for learning logical theories, which has hardly received any attention

Learning and Reasoning with Events for Language Understanding Commonsense (background) knowledge plays a crucial role in our understanding, which generally "pop up" as supplementary assumptions or expectations and can be used "on the fly" when performing reasoning at the context-level of a text document. The majority of existing approaches of language understanding mainly focuses on performing low-level forms of reasoning at the sentence level to achieve tasks. However, if we want to move forward to more robust AI systems, we need high-level reasoning abilities that combine different knowledge in a principled way. Reasoning about textual documents is event-centric, which intrinsically relies on the events affordances (aspects or expectations of the events), and the interactions (e.g. causality relations) between them. Future work will concern developing methods for reasoning about events in language understanding.

References

BIBLIOGRAPHY

Rana Alshaikh, Zied Bouraoui, and Steven Schockaert. Learning conceptual spaces with disentangled facets. In *CoNLL*, pages 131–139, 2019.

Rana Alshaikh, Zied Bouraoui, Shelan S. Jeawak, and Steven Schockaert. A mixture-of-experts model for learning multi-facet entity embeddings. In *COLING*, pages 5124–5135, 2020.

Rana Alshaikh, Zied Bouraoui, and Steven Schockaert. Hierarchical linear disentanglement of data-driven conceptual spaces. In *IJCAI*, pages 3573–3579, 2020.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. Generalizing and improving bilingual word embedding mappings with a multi-step framework of linear transformations. In *Proc. AAAI*, pages 5012–5019, 2018.

Islam Beltagy, Cuong Chau, Gemma Boleda, Dan Garrette, Katrin Erk, and Raymond Mooney. Montague meets Markov: Deep semantics with probabilistic logical form. In *Second Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 11–21, USA, jun 2013. Association for Computational Linguistics.

Salem Benferhat and Zied Bouraoui. Min-based possibilistic dl-lite. *J. Log. Comput.*, 27(1):261–297, 2017.

Salem Benferhat, Zied Bouraoui, Odile Papini, and Eric Würbel. Assertional removed sets merging of dl-lite knowledge bases. In Nahla Ben Amor, Benjamin Quost, and Martin Theobald, editors, *Scalable Uncertainty Management - 13th International Conference, SUM 2019, Compiègne, France, December 16-18, 2019, Proceedings*, volume 11940 of *Lecture Notes in Computer Science*, pages 207–220, Compiègne, France., 2019. Springer.

Leopoldo E. Bertossi. *Database Repairing and Consistent Query Answering*. Synthesis Lectures on Data Management. Morgan & Claypool Publishers, 2011.

Sumithra Bhakthavatsalam, Chloe Anastasiades, and Peter Clark. GenericsKB: A knowledge base of generic statements. *CoRR*, abs/2005.00660, 2020.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, 2003.

Rishi Bommasani, Kelly Davis, and Claire Cardie. Interpreting pretrained contextualized representations via reductions to static embeddings. In *Proceedings ACL*, pages 4758–4781, 2020.

Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *NIPS*, pages 2787–2795, 2013.

Stefan Borgwardt, İsmail İlkan Ceylan, and Thomas Lukasiewicz. Recent advances in querying probabilistic knowledge bases. In *IJCAI*, pages 5420–5426, 2018.

Zied Bouraoui and Steven Schockaert. Learning conceptual space representations of interrelated concepts. In Jérôme Lang, editor, *IJCAI*, pages 1760–1766, 2018.

Zied Bouraoui and Steven Schockaert. Automated rule base completion as bayesian concept induction. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 6228–6235. AAAI Press, 2019.

Zied Bouraoui, Shoaib Jameel, and Steven Schockaert. Inductive reasoning about ontologies using conceptual spaces. In *AAAI*, pages 4364–4370, 2017.

Zied Bouraoui, Shoaib Jameel, and Steven Schockaert. Relation induction in word embeddings revisited. In Emily M. Bender, Leon Derczynski, and Pierre Isabelle, editors, *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 1627–1637. Association for Computational Linguistics, 2018.

Zied Bouraoui, José Camacho-Collados, Luis Espinosa Anke, and Steven Schockaert. Modelling semantic categories using conceptual neighborhood. In *AAAI*, pages 7448–7455. AAAI Press, 2020.

Zied Bouraoui, José Camacho-Collados, and Steven Schockaert. Inducing relational knowledge from BERT. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 7456–7463. AAAI Press, 2020.

Zied Bouraoui, Sébastien Konieczny, Truong-Thanh Ma, and Ivan Varzinczak. Model-based merging of open-domain ontologies. In *32nd IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2020, Baltimore, MD, USA, November 9-11, 2020*, pages 29–34. IEEE, 2020.

Zied Bouraoui, Jean-Marie Lagniez, Pierre Marquis, and Valentin Montmirail. Consolidating modal knowledge bases. In Giuseppe De Giacomo, Alejandro Catalá, Bistra Dilkina, Michela Milano, Senén Barro, Alberto Bugarín, and Jérôme Lang, editors, *ECAI 2020 - 24th European Conference on Artificial Intelligence, 29 August-8 September 2020, Santiago de Compostela, Spain, August 29 - September 8, 2020 - Including 10th Conference on Prestigious Applications of Artificial Intelligence (PAIS 2020)*, volume 325 of *Frontiers in Artificial Intelligence and Applications*, pages 648–655. IOS Press, 2020.

Zied Bouraoui, Sébastien Konieczny, Thanh Ma, and Ivan Varzinczak. Tree edit distance based ontology merging evaluation framework. In Gérard Memmi, Baijian Yang, Linghe Kong, Tianwei Zhang, and Meikang Qiu, editors, *Knowledge Science, Engineering and Management - 15th International Conference, KSEM 2022, Singapore, August 6-8, 2022, Proceedings, Part II*, volume 13369 of *Lecture Notes in Computer Science*, pages 383–395. Springer, 2022.

Zied Bouraoui, Sébastien Konieczny, Truong-Thanh Ma Nicolas Schwind, and Ivan Varzinczak. Region-based merging of open-domain terminological knowledge. In *Proceedings of the 19th International Conference on Principles of Knowledge Representation and Reasoning, KR 2022*, 2022.

Xingyu Cai, Jiaji Huang, Yuchen Bian, and Kenneth Church. Isotropy in the contextual embedding space: Clusters and manifolds. In *International Conference on Learning Representations*, 2020.

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

Zhao-Min Chen, Xiu-Shen Wei, Peng Wang, and Yanwen Guo. Multi-label image recognition with graph convolutional networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5177–5186, 2019.

Zheng Chen, Sun Yu, Wan Shengxian, and Yu Dianhai. RLTM: An Efficient Neural IR Framework for Long Documents. In *the proceeding of International Joint Conference on Artificial Intelligence (IJCAI'19)*, pages 5457–5463, Macao, China, 2019. ijcai.org.

Massimiliano Ciaramita and Mark Johnson. Supersense tagging of unknown nouns in WordNet. In *Proc. EMNLP*, pages 168–175, 2003.

Michael AA Cox and Trevor F Cox. Multidimensional scaling. In *Handbook of data visualization*, pages 315–347. Springer, 2008.

Rajarshi Das, Manzil Zaheer, and Chris Dyer. Gaussian LDA for topic models with word embeddings. In *Proceedings ACL*, pages 795–804, 2015.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proc. NAACL-HLT*, 2019.

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

Aleksandr Drozd, Anna Gladkova, and Satoshi Matsuoka. Word embeddings, analogies, and machine learning: Beyond king - man + woman = queen. In *Proceedings of the 26th International Conference on Computational Linguistics*, pages 3519–3530, 2016.

Didier Dubois, Henri Prade, Francesc Esteva, Pere Garcia, and Lluis Godo. A logical approach to interpolation based on similarity relations. *International Journal of Approximate Reasoning*, 17(1):1–36, 1997.

Maryam Fazel. *Matrix rank minimization with applications*. PhD thesis, Stanford University, 2002.

Peter Gärdenfors and Mary-Anne Williams. Reasoning about categories in conceptual spaces. In *IJCAI*, pages 385–392, 2001.

Peter Gärdenfors. *Conceptual spaces: The geometry of thought*. MIT press, 2000.

Peter Gärdenfors. How to make the semantic web more semantic. In A.C. Varzi and L. Vieu, editors, *Formal Ontology in Information Systems*, pages 19–36. IOS Press, 2004.

Anna Gladkova, Aleksandr Drozd, and Satoshi Matsuoka. Analogy-based detection of morphological and semantic relations with word embeddings: what works and what doesn't. In *Proc. NAACL Student Research Workshop*, pages 8–15, 2016.

Robert L Goldstone. Isolated and interrelated concepts. *Memory & Cognition*, 24(5):608–628, 1996.

Víctor Gutiérrez-Basulto, Jean Christoph Jung, Carsten Lutz, and Lutz Schröder. Probabilistic description logics for subjective uncertainty. *J. Artif. Intell. Res.*, 58:1–66, 2017.

Patrick Hohenecker and Thomas Lukasiewicz. Ontology Reasoning with Deep Neural Networks. In *the proceeding of International Joint Conference on Artificial Intelligence (IJCAI-20)*, pages 5060–5064, Yokohama, Japan, 7 2020. ijcai.org.

Timo Homburg, Steffen Staab, and Daniel Janke. Geosparql+: Syntax, semantics and system for integrated querying of graph, raster and vector data. In *the proceeding of the International Semantic Web Conference (ISWC'2020)*, pages 258–275, Athens, Greece, 2020. CEUR-WS.org.

Anthony Hunter, Simon Parsons, and Michael J. Wooldridge. Measuring inconsistency in multi-agent systems. *KI*, 28(3):169–178, 2014.

Gerhard Jäger. Natural color categories are convex sets. In *17th Amsterdam Colloquium on Logic, Language and Meaning*, pages 11–20, 2009.

Shoaib Jameel, Zied Bouraoui, and Steven Schockaert. Member: Max-margin based embeddings for entity retrieval. In Noriko Kando, Tetsuya Sakai, Hideo Joho, Hang Li, Arjen P. de Vries, and Ryen W. White, editors, *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7-11, 2017*, pages 783–792. ACM, 2017.

Shoaib Jameel, Zied Bouraoui, and Steven Schockaert. Unsupervised learning of distributional relation vectors. In Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of*

the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers, pages 23–33. Association for Computational Linguistics, 2018.

Prodromos Kolyvakis, Alexandros Kalousis, and Dimitris Kiritsis. Deepalignment: Unsupervised ontology matching with refined word vectors. In *Proc. NAACL-HLT*, pages 787–798, 2018.

Sri Krishna Kumar and Jennifer A Harding. Description logic–based knowledge merging for concrete- and fuzzy-domain ontologies. *Journal of Engineering Manufacture*, 230(5):954–971, 2016.

Na Li, Zied Bouraoui, and Steven Schockaert. Ontology completion using graph convolutional networks. In Chiara Ghidini, Olaf Hartig, Maria Maleshkova, Vojtech Svátek, Isabel F. Cruz, Aidan Hogan, Jie Song, Maxime Lefrançois, and Fabien Gandon, editors, *The Semantic Web - ISWC 2019 - 18th International Semantic Web Conference, Auckland, New Zealand, October 26-30, 2019, Proceedings, Part I*, volume 11778 of *Lecture Notes in Computer Science*, pages 435–452. Springer, 2019.

Aoxue Li, Weiran Huang, Xu Lan, Jiashi Feng, Zhenguo Li, and Liwei Wang. Boosting few-shot learning with adaptive margin loss. In *Proc. CVPR*, pages 12573–12581, 2020.

Na Li, Zied Bouraoui, José Camacho-Collados, Luis Espinosa Anke, Qing Gu, and Steven Schockaert. Modelling general properties of nouns by selectively averaging contextualised embeddings. In Zhi-Hua Zhou, editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*, pages 3850–3856. ijcai.org, 2021.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019.

Ken McRae et al. Semantic feature production norms for a large set of living and nonliving things. *Behavior research methods*, 37:547–559, 2005.

Tomás Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In Lucy Vanderwende, Hal Daumé III, and Katrin Kirchhoff, editors, *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA*, pages 746–751. The Association for Computational Linguistics, 2013.

George A Miller. Wordnet: a lexical database for English. *Communications of the ACM*, 38(11):39–41, 1995.

Rym Mohamed, Zied Loukil, and Zied Bouraoui. Qualitative-based possibilistic *EL* ontology. In Tim Miller, Nir Oren, Yuko Sakurai, Itsuki Noda, Bastin Tony Roy Savarimuthu, and Tran Cao Son, editors, *PRIMA 2018: Principles and Practice of Multi-Agent Systems - 21st International Conference, Tokyo, Japan, October 29 - November 2, 2018, Proceedings*, volume 11224 of *Lecture Notes in Computer Science*, pages 552–559. Springer, 2018.

Rim Mohamed, Zied Loukil, Faïez Gargouri, and Zied Bouraoui. Evolution of prioritized el ontologies. In *Advances and Trends in Artificial Intelligence. Theory and Practices in Artificial Intelligence - 35th International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2022, Kitakyushu, Japan, July 19-22, 2022, Proceedings*, volume 13343 of *Lecture Notes in Computer Science*, pages 859–870. Springer, 2022.

Rym Mohamed, Zied Loukil, Faiz Gargouri, and Zied Bouraoui. Conflict-base inconsistency tolerent query answering. In *Proceedings of the International Florida Artificial Intelligence Research Society Conference, FLAIRS 2022*. AAAI Press, 2022.

Rym Mohamed, Zied Loukil, Faiz Gargouri, and Zied Bouraoui. Min-based conditioning of possibilistic el ontology. In *Proceedings of the International Florida Artificial Intelligence Research Society Conference, FLAIRS 2022*. AAAI Press, 2022.

Roberto Navigli and Simone Paolo Ponzetto. Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250, 2012.

Fedor Nikolaev and Alexander Kotov. Joint word and entity embeddings for entity retrieval from a knowledge graph. In *Proc. ECIR*, pages 141–155, 2020.

Robert M Nosofsky. Choice, similarity, and the context theory of classification. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 10(1):104–114, 1984.

Kezban Dilek Onal, Ye Zhang, Ismail Sengor Altingovde, Md Mustafizur Rahman, Pinar Karagoz, Alex Braylan, Brandon Dang, Heng-Lu Chang, Henna Kim, Quinten McNamara, et al. Neural information retrieval: At the end of the early years. *Information Retrieval Journal*, 21(2-3):111–182, 2018.

Daniel N Osherson, Edward E Smith, Ormond Wilkie, Alejandro Lopez, and Eldar Shafir. Category-based induction. *Psychological Review*, 97(2):185–200, 1990.

Matías Osta-Vélez and Peter Gärdenfors. Category-based induction in conceptual spaces. *Journal of Mathematical Psychology*, 96, 2020.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. GloVe: Global vectors for word representation. In *EMNLP*, pages 1532–1543, 2014.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *Technical Report*, 2019.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

Lance J Rips. Inductive judgments about natural categories. *Journal of Verbal Learning and Verbal Behavior*, 14(6):665–681, 1975.

Tim Rocktäschel and Sebastian Riedel. End-to-end differentiable proving. In *NIPS*, pages 3788–3800, 2017.

Marco Rospocher and Francesco Corcoglioniti. Joint Posterior Revision of NLP Annotations via Ontological Knowledge. In *the proceeding of International Joint Conference on Artificial Intelligence (IJCAI-18)*, pages 4316–4322, Stockholm, Sweden, 7 2018. ijcai.org.

Steven Schockaert and Sanjiang Li. Combining RCC5 relations with betweenness information. In Francesca Rossi, editor, *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*, pages 1083–1089, Beijing, China, 2013. IJCAI/AAAI.

Steven Schockaert and Henri Prade. Interpolative and extrapolative reasoning in propositional theories using qualitative knowledge about conceptual spaces. *Artif. Intell.*, 202:86–131, 2013.

Steven A Sloman. Feature-based induction. *Cognitive Psychology*, 25(2):231–280, 1993.

Jake Snell, Kevin Swersky, and Richard S. Zemel. Prototypical networks for few-shot learning. In *Proc. NIPS*, pages 4077–4087, 2017.

Richard Socher, Milind Ganjoo, Christopher D Manning, and Andrew Ng. Zero-shot learning through cross-modal transfer. In *Proc. NIPS*, pages 935–943, 2013.

Robert Speer, Catherine Havasi, and Henry Lieberman. Analogyspace: reducing the dimensionality of common sense knowledge. In *Proc. AAAI*, pages 548–553, 2008.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. RotatE: Knowledge graph embedding by relational rotation in complex space. In *ICLR*, 2019.

Thomas Pellissier Tanon, Denny Vrandecic, Sebastian Schaffert, Thomas Steiner, and Lydia Pintscher. From freebase to wikidata: The great migration. In *International World Wide Web Conference (WWW'2016)*, pages 1419–1428, Switzerland, 2016. World Wide Web Conference.

William Timkey and Marten van Schijndel. All bark and no bite: Rogue dimensions in transformer language models obscure representational quality. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4527–4546, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

Théo Trouillon, Christopher R. Dance, Éric Gaussier, Johannes Welbl, Sebastian Riedel, and Guillaume Bouchard. Knowledge graph completion via complex tensor factorization. *J. Mach. Learn. Res.*, 18:130:1–130:38, 2017.

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

Manuel Vimercati, Federico Bianchi, Mauricio Soto, and Matteo Palmonari. Mapping lexical knowledge to distributed models for ontology concept invention. In *Proc. IA*AI*, pages 572–587, 2019.

Ivan Vulic, Edoardo Maria Ponti, Robert Litschko, Goran Glavas, and Anna Korhonen. Probing pretrained language models for lexical semantics. In *Proceedings EMNLP*, pages 7222–7240, 2020.

Z. Wang, K. Wang, Y. Jin, and G. Qi. Ontomerge: A system for merging DL-Lite ontologies. *CEUR Workshop Proceedings*, 969:16–27, 01 2012.

Jiang Wang, Yi Yang, Junhua Mao, Zhiheng Huang, Chang Huang, and Wei Xu. Cnn-rnn: A unified framework for multi-label image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2285–2294, 2016.

Zhouxia Wang, Tianshui Chen, Guanbin Li, Ruijia Xu, and Liang Lin. Multi-label image recognition by recurrently discovering attentional regions. In *Proceedings of the IEEE international conference on computer vision*, pages 464–472, 2017.

Yixiao Wang, Zied Bouraoui, Luis Espinosa Anke, and Steven Schockaert. Deriving word vectors from contextualized language models using topic-aware mention selection. In *Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP-2021)*, pages 185–194, Online, August 2021. Association for Computational Linguistics.

Yixiao Wang, Zied Bouraoui, Luis Espinosa Anke, and Steven Schockaert. Sentence selection strategies for distilling word embeddings from bert. In *Proceedings of The 14th Language Resources and Evaluation Conference, LREC 2022*. European Language Resources Association, 2022.

Chen Xing, Negar Rostamzadeh, Boris N. Oreshkin, and Pedro O. Pinheiro. Adaptive cross-modal few-shot learning. In *Proc. NIPS*, pages 4848–4858, 2019.

Kun Yan, Zied Bouraoui, Ping Wang, Shoaib Jameel, and Steven Schockaert. Aligning visual prototypes with BERT embeddings for few-shot learning. In Wen-Huang Cheng, Mohan S. Kankanhalli, Meng Wang, Wei-Ta Chu, Jiaying Liu, and Marcel Worring, editors, *ICMR '21: International Conference on Multimedia Retrieval, Taipei, Taiwan, August 21-24, 2021*, pages 367–375. ACM, 2021.

Kun Yan, Zied Bouraoui, Ping Wang, Shoaib Jameel, and Steven Schockaert. Few-shot image classification with multi-facet prototypes. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2021, Toronto, ON, Canada, June 6-11, 2021*, pages 1740–1744. IEEE, 2021.

Kun Yan, Chenbin Zhang, Jun Hou, Ping Wang, Zied Bouraoui, Shoaib Jameel, and Steven Schockaert. Inferring prototypes for multi-label few-shot image classification with word vector guided attention. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 2991–2999. AAAI Press, 2022.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. In *ICLR*, 2015.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. XLNet: Generalized autoregressive pretraining for language understanding. *arXiv:1906.08237*, 2019.

Vacit Oguz Yazici, Abel Gonzalez-Garcia, Arnau Ramisa, Bartlomiej Twardowski, and Joost van de Weijer. Orderless recurrent models for multi-label classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13440–13449, 2020.