Distilling Semantic Concept Embeddings from Contrastively Fine-Tuned Language Models

Na Li
University of Shanghai for Science and Technology
China
li_na@usst.edu.cn

Hanane Kteich
CRIL CNRS & University of Artois
France
kteich@cril.fr

Zied Bouraoui
CRIL CNRS & University of Artois
France
bouraoui@cril.fr

Steven Schockaert
Cardiff University
United Kingdom
schockaerts1@cardiff.ac.uk

ABSTRACT

Learning vectors that capture the meaning of concepts remains a fundamental challenge. Somewhat surprisingly, perhaps, pre-trained language models have thus far only enabled modest improvements to the quality of such *concept embeddings*. Current strategies for using language models typically represent a concept by averaging the contextualised representations of its mentions in some corpus. This is potentially sub-optimal for at least two reasons. First, contextualised word vectors have an unusual geometry, which hampers downstream tasks. Second, concept embeddings should capture the semantic properties of concepts, whereas contextualised word vectors are also affected by other factors. To address these issues, we propose two contrastive learning strategies, based on the view that whenever two sentences reveal similar properties, the corresponding contextualised vectors should also be similar. One strategy is fully unsupervised, estimating the properties which are expressed in a sentence from the neighbourhood structure of the contextualised word embeddings. The second strategy instead relies on a distant supervision signal from ConceptNet. Our experimental results show that the resulting vectors substantially outperform existing concept embeddings in predicting the semantic properties of concepts, with the ConceptNet-based strategy achieving the best results. These findings are furthermore confirmed in a clustering task and in the downstream task of ontology completion.

CCS CONCEPTS

• **Computing methodologies** → **Lexical semantics**; **Ontology engineering**; **Unsupervised learning**.

KEYWORDS

word embedding, language models, contrastive learning, commonsense knowledge

SIGIR '23, July 23–27, 2023, Taipei, Taiwan.

ACM ISBN 978-1-4503-9408-6/23/07…$15.00
https://doi.org/10.1145/3539618.3591667

ACM Reference Format:
Na Li, Hanane Kteich, Zied Bouraoui, and Steven Schockaert. 2023. Distilling Semantic Concept Embeddings from Contrastively Fine-Tuned Language Models. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '23), July 23–27, 2023, Taipei, Taiwan.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3539618.3591667

1 INTRODUCTION

Since the introduction of BERT [10], the focus in Natural Language Processing (NLP) has been on fine-tuning and exploiting large pre-trained language models, especially for solving sentence and paragraph level tasks. However, accurately modelling the meaning of individual words, in the form of static (i.e. not contextualised) vectors, also continues to be an important challenge. Static word vectors are used, among others, as pre-trained label embeddings for zero-shot [38, 48] and few-shot learning [23, 29, 58–61]; as concept representations for ontology alignment [27], ontology completion [32] and taxonomy learning [39, 53]; for lexical substitution [56] and topic modelling [9, 11, 63]; and for analysing social biases [6]. Motivated by such applications, this paper focuses on representations of *concepts*, rather than named entities.

The distributional hypothesis [13, 20] suggests that the meaning of a concept can be inferred from the contexts in which it appears. Standard word embedding models [42, 47] implement this idea by using bag-of-words representations of these contexts. Clearly, such representations can only capture what is revealed about a concept in a very approximate way. Pre-trained language models (LMs), on the other hand, are able to capture meaning at the sentence level. LMs should thus enable us to obtain higher-quality context representations, which we would expect to translate into higher-quality concept embeddings. In particular, several authors have explored the idea that embeddings of concepts can be obtained by aggregating the contextualised embeddings of their mentions in some corpus [6, 12, 16, 19, 31, 55, 57]. While improvements over standard word embeddings are routinely reported, such improvements tend to be relatively small, and they are not always consistent.

There are at least two challenges when it comes to learning concept embeddings in this way. First, contextualised word vectors are highly anisotropic [12]. For unsupervised sentence embeddings, strategies aimed at reducing anisotropy have been found to result in substantial performance gains [24, 30, 33]. We may thus expect

that concept embeddings can similarly benefit from such strategies. Second, and more fundamentally, contextualised word vectors do not only capture information about the meaning of words but also about their syntactic role and other characteristics of the sentences in which they appear [22, 37, 41, 50, 51]. If we are interested in modelling the meaning of concepts, it thus seems beneficial to specialise the contextualised word vectors towards this aspect. Ideally, two contextualised word vectors should be similar if the corresponding sentences express similar properties, and dissimilar otherwise. This key idea is illustrated in the following example.

Example 1. *Consider the following sentences*[1]*:*

(i) Submarines can hide under the water.
(ii) Some submarines run on diesel engines.
(iii) Some sharks live at the bottom of deep underwater canyons.
(iv) Trucks are used to transport people or things, they use fuel known as diesel.

We would like the contextualised representation of submarines *in sentence (i) to be similar to the contextualised representation of* sharks *in sentence (iii), as both sentences assert that the target concept has the property of being underwater. Similarly, we would like the representation of* submarines *in sentence (ii) to be similar to the representation of* trucks *in sentence (iv).*

If we are able to learn contextualised word vectors that focus on the semantic properties that are expressed in a given sentence, we should be able to learn high-quality concept embeddings by averaging these contextualised representations across different sentences.

In this paper, we propose and analyse a number of strategies based on contrastive learning to address the two aforementioned issues. Contrastive learning has already been successfully used for alleviating the anisotropy of BERT-based word and sentence embeddings [17, 33], based on the idea that embeddings of corrupted inputs should be similar to embeddings of the original word or sentence. Different from these approaches, our motivation for using contrastive learning is to move contextualised word vectors that capture similar semantic properties closer together, while vectors capturing different properties are pushed further apart.

Crucially, to implement this idea, we need examples of sentences that express similar properties. We propose two strategies for identifying such sentences. Our first strategy is purely unsupervised. The main idea is to rely on the neighbourhood structure of standard contextualised word vectors. First note that when obtaining contextualised word vectors, we mask the target concept, following [31]. This ensures that contextualised word vectors reflect the sentence context of the given concept, rather than any prior knowledge about the concept that is captured by the language model itself. Now suppose we have a contextualised representation of *submarine*, and we look for the most similar contextualised word vectors, across a given corpus. Since the target concept is masked, these vectors may correspond to different words. Suppose, for instance, that they correspond to the words *car*, *truck* and *airplane*. Then we can intuitively assume that the given sentence expresses the property of being a vehicle. Based on this idea, we can identify sentences that are likely to express the same property. Our second strategy uses a form of distant supervision, using knowledge about the commonsense properties of concepts from ConceptNet [49]. For example, ConceptNet contains the triple *(gun,HasProperty,dangerous)*. Given this triple, if a sentence contains both the words *gun* and *dangerous*, we assume it expresses that guns are dangerous. For each property encoded in ConceptNet, we can thus find sentences which express that the target concept has that property. This, in particular, allows us to find sentences that express the same property.

We experimentally compare the concept embeddings that are obtained with the two aforementioned strategies. We are specifically interested in the extent to which different kinds of semantic properties can be predicted from these embeddings. We also evaluate our embeddings in a clustering task and an ontology completion task [8, 32]. For both strategies, we find that our concept embeddings consistently outperform existing models by a substantial margin.

2 RELATED WORK

The use of pre-trained language models for generating static word embeddings has already been extensively explored. A popular strategy is to aggregate the contextualised representation of a word w across a number of sentences mentioning this word [6, 12, 55]. Several variations of this strategy have been studied, which mostly differ in how the contextualised representation of w is computed. It is common to use the representation from the final layer of the transformer model or to average the representations from the final four layers, while Vulić et al. [55] suggested averaging the first k layers, with the optimal k depending on the task. For words that consist of multiple tokens, the representations of these tokens are typically averaged. To aggregate the contextualised representations of a given word w across multiple sentences, the most common strategy is to simply average them. Ethayarajh [12] instead proposed to take the first principal component, which produces almost the same result, given that the contextualised vectors are all located in a very narrow cone. In this paper, we build on the approach from Li et al. [31], which masks the target word w and uses the contextualised representation of the mask token; this approach is discussed in more detail in the next section. Beyond averaging-based strategies, some approaches have been inspired by Word2Vec [42] or GloVe [47], relying on BERT to obtain context embeddings [19, 57], or to generate synthetic co-occurrence counts [16].

Instead of relying on words in context, some approaches simply feed the word w to the language model. Bommasani et al. [6] found this to perform poorly with pre-trained models. However, better results were reported by Vulić et al. [54], after fine-tuning the BERT encoder on synonymy and antonymy pairs. Gajbhiye et al. [15] jointly fine-tuned a BERT encoder for concepts and an encoder for properties, using hypernyms from Microsoft Concept Graph [25] and sentences from GenericsKB [5] as training data. MirrorBERT [33] is a BERT encoder for both words and sentences, which is trained in a fully self-supervised way. It uses dropout to generate different variants of the same input, and then fine-tunes BERT such that these variants are closer to each other than to encodings of other inputs. The resulting encoder can generate high-quality word vectors, again without needing sentences mentioning the word in context. MirrorWiC [34] can be seen as an adaptation of the MirrorBERT strategy to words in context. In particular, given a

[1] All sentences were taken from GenericsKB [5].

sentence s mentioning some word w, multiple encodings of w are obtained by (i) randomly masking different spans in s and (ii) using dropout. The model then encourages different encodings of same sentence to be closer to each other than to encodings obtained from different sentences (even if the target word w is the same).

The aforementioned approaches have been developed with different tasks in mind. While word similarity benchmarks remain a popular choice for evaluating word vectors, Li et al. [31] and Gajbhiye et al. [15] were specifically interested in predicting the commonsense properties of concepts, while Liu et al. [34] focused on word sense disambiguation. Accordingly, some of these approaches have complementary strengths. For instance, the model from Li et al. [31] outperformed the baselines on concept categorisation tasks, but under-performed in word similarity. In terms of downstream applications, since the introduction of BERT, word embeddings have primarily been used in settings where word meaning has to be modelled in the absence of any sentence context. For instance, word embeddings have been used to estimate class prototypes for few-shot learning, e.g. in image classification [58, 60, 61] and for slot tagging in dialogue systems [23]. In [29], word vector similarity was used to set an adaptive margin, as part of a margin-based model for few-shot image classification, to capture the idea that image classes with similar labels can be harder to differentiate. Word embeddings have also been used for modelling label dependencies in multi-label classification [59]. Furthermore, word vectors have been used for ontology engineering tasks, e.g. for aligning ontologies [27] or for inferring plausible rules [32]. In such applications, what matters is that concepts with similar word vectors have similar properties. We will focus on ontology completion in more detail in Section 5.3. In other applications, what matters is rather that clusters of word vectors are semantically coherent, e.g. when using word vectors for learning taxonomies [39, 53] or for topic modelling [9, 11, 63]. Word vectors are much easier to train than language models, and can thus more easily be adapted. This advantage has been exploited to learn personal word embeddings, as part of a system for personalised search [62], or for studying how word meaning changes over time [28]. Finally, some authors have found that even for tasks where we need to model the meaning of words in context, using static word vectors can sometimes be beneficial [1, 35, 56].

3 DISTILLING CONCEPT EMBEDDINGS

In this section, we recall the concept embedding strategy from Li et al. [31], which uses a pre-trained BERT model. The aim of our paper is to analyse how better concept embeddings can be obtained by instead relying on a suitably fine-tuned BERT model. Our proposed fine-tuning strategies will be the focus of Section 4.

Let $s_1, ..., s_n$ be sentences in which some concept c is mentioned. To obtain a vector representation of c from the sentence s_i, Li et al. [31] replace c by the *<mask>* token and take the final-layer contextualised representation of this token, using a BERT-based language model. By masking the concept c, the resulting vector intuitively captures what the sentence s_i reveals about the meaning of c, rather than any prior knowledge about the meaning of c that is encoded in the language model itself. They found that this masking strategy improves how well the resulting embeddings capture the semantic properties of concepts. Let $\mathbf{x_1}, ..., \mathbf{x_n}$ be the vectors that are thus obtained from the available sentences. We refer to these vectors as the *mention vectors* of concept c. We write $\mu(c) = \{\mathbf{x_1}, ..., \mathbf{x_n}\}$ for the set of mention vectors associated with c. An embedding of concept c can be obtained by averaging these mention vectors:

$$\mathbf{c} = \frac{1}{|\mu(c)|} \sum \{\mathbf{x} \mid \mathbf{x} \in \mu(c)\}$$

However, not all sentences are equally informative. Li et al. [31] in particular highlighted issues that arise when sentences use concepts in idiosyncratic ways. For instance, sentences about the children's song "Mary had a little lamb" are unlikely to be useful for learning a representation of the concept *lamb*. To reduce the impact of such idiosyncratic sentences, they proposed the following filtering strategy. Let V be a vocabulary of concepts and let $M = \bigcup_{v \in V} \mu(v)$ be the set of all mention vectors, across all words in the vocabulary. For each mention vector $\mathbf{x}$ in $\mu(c)$, we compute its k nearest neighbours among the vectors in M. If all k of these neighbours belong to $\mu(c)$, $\mathbf{x}$ is deemed to be idiosyncratic. The embedding of concept c is then obtained by averaging the remaining mention vectors, after removing the idiosyncratic ones. The underlying intuition is based on the idea that the mention vectors in $\mu(c)$ capture the properties of c. If all the neighbours of such a mention vector $\mathbf{x}$ are associated with c, it suggests that the property which is captured by $\mathbf{x}$ only applies to that concept and is thus unlikely to be important.

4 CONTRASTIVE LEARNING STRATEGIES

Each mention vector in $\mu(c)$ intuitively encodes what the corresponding sentence reveals about the concept c. It would thus be desirable if two mention vectors were similar if and only if the corresponding sentences reveal similar properties. Unfortunately, this is not always the case, given that contextualised vectors are affected by aspects such as word position, word frequency, and punctuation [37, 41, 51], which are irrelevant to word meaning, as well as the syntactic role of a word [22, 50], which is only loosely related. Our solution is to fine-tune the mention vectors using a contrastive learning strategy. While contrastive learning is a popular representation learning technique, it is usually applied in a purely unsupervised setting. For instance, to learn sentence embeddings using contrastive learning, one usually trains the model such that embeddings of corrupted versions of the same sentence are similar to each other, and dissimilar from embeddings of other sentences [17, 33]. The same strategy has been used in [33] for obtaining word embeddings from BERT. While it leads to embeddings that perform well on word similarity benchmarks, as we will see in our experiments, they are less suitable for tasks such as ontology completion, where we need concept embeddings that capture the semantic properties of the corresponding concepts.

In contrast to these existing approaches, our strategies will rely on weakly labelled training examples. Each example consists of two sentence-concept pairs, (s_1, c_1) and (s_2, c_2), where c_i is a concept that is mentioned in sentence s_i. For positive training examples, the assumption is that the property that sentence s_1 expresses about concept c_1 is the same as what sentence s_2 expresses about c_2. For instance, if we write $s_{(i)}$ for sentence (i) from Example 1, and similar for $s_{(ii)}$ and $s_{(iii)}$, then (s_i, *submarines*); (s_{iii}, *sharks*) could be a positive training example, while (s_i, *submarines*); (s_{ii}, *submarines*) could be a negative example. To implement our strategy, we thus

first need to find a way to obtain such weakly labelled training examples. In Section 4.1 we propose two solutions for this problem: an unsupervised strategy which relies on the neighbourhood structure of the mention vectors, and a distantly supervised strategy which is based on ConceptNet. In Section 4.2 we then describe how the resulting training examples can be used for fine-tuning the model.

4.1 Constructing Weakly Labelled Examples

We propose two strategies for obtaining weakly labelled training examples. These examples will then be used in Section 4.2 for fine-tuning the mention vectors.

4.1.1 Neighbourhood Structure. Consider sentences (i) and (iii) from Example 1. Even though these sentences express a similar property (i.e. being located under water), the resulting mention vectors are not actually similar, even after masking the target concepts. In fact, this is precisely our motivation for fine-tuning the mention vectors. To discover sentences which are likely to express a similar property, it is thus not sufficient to directly compare the corresponding mention vectors. Let us write $\phi(s, c)$ for the mention vector which is obtained after masking concept c in sentence s. Essentially, two mention vectors $\phi(s_1, c_1)$ and $\phi(s_2, c_2)$ are similar if the following two conditions are satisfied for the sentences s_1 and s_2: (i) they express a similar property about their target concepts (i.e. c_1 and c_2) and (ii) they have a similar structure, with c_1 and c_2 moreover occurring in a similar syntactic role. In particular, if two mention vectors are similar, it is likely that they capture a similar property, even if the converse is not true. This insight can be used to compare the mention vectors $\phi(s_1, c_1)$ and $\phi(s_2, c_2)$ in an indirect way: we obtain the set X_1 of mentions vectors which are most similar to $\phi(s_1, c_1)$ and the set X_2 of mention vectors which are most similar to $\phi(s_2, c_2)$. If the concepts associated with the mention vectors in X_1 are broadly the same as the concepts associated with the mention vectors in X_2, it intuitively means that the property expressed by the vector $\phi(s_1, c_1)$ applies to the same set of concepts as the property expressed by the vector $\phi(s_2, c_2)$. In such a case, it is likely that $\phi(s_1, c_1)$ and $\phi(s_2, c_2)$ express the same property.

We now describe the proposed method more formally. Let V be the vocabulary of all concepts and let $M = \bigcup_{c \in V} \mu(c)$ be the set of available mention vectors. In the following, we will assume that $\mu(c) \cap \mu(d) = \emptyset$ for $c \neq d$, i.e. we never have the exact same mention vector for different concepts. This assumption simplifies the formulations and is satisfied in practice. In particular, we can then link each mention vector $\mathbf{x} \in M$ to its unique corresponding concept, which we denote by $\omega(\mathbf{x})$, i.e. we have $\omega(\mathbf{x}) = c$ iff $\mathbf{x} \in \mu(c)$. For a mention vector $\mathbf{x} \in M$, we write $neigh(\mathbf{x})$ for its k nearest neighbours from M, in terms of cosine similarity. Our central assumption is that when two mention vectors $\mathbf{x}$ and $\mathbf{y}$ express a similar property, then the concepts associated with the mention vectors in $neigh(\mathbf{x})$ and $neigh(\mathbf{y})$ will be similar. Formally, we define the compatibility degree $\pi(\mathbf{x}, \mathbf{y})$ between $\mathbf{x}$ and $\mathbf{y}$ as follows:

$$\frac{\sum_{c \in V} \min(freq(c, neigh(\mathbf{x})), freq(c, neigh(\mathbf{y})))}{\sum_{c \in V} \max(freq(c, neigh(\mathbf{x})), freq(c, neigh(\mathbf{y})))}$$

where $freq(c, X) = |\{\mathbf{x} \in X : \omega(\mathbf{x}) = c\}|$ is the number of mention vectors in X that are associated with concept c. The following toy example provides an illustration of how $\pi(\mathbf{x}, \mathbf{y})$ is computed.

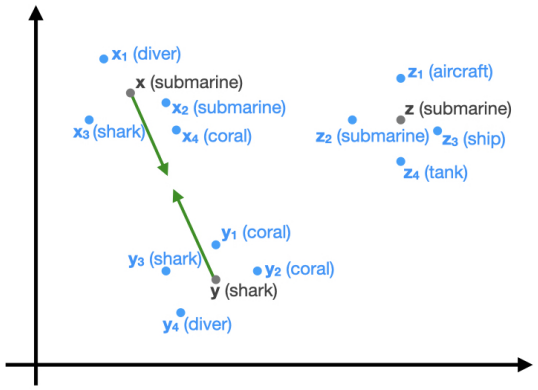

Figure 1: Illustration of the neighbourhood-based selection of positive examples.

Example 2. *Figure 1 focuses on mention vectors* $\mathbf{x}$, $\mathbf{y}$ *and* $\mathbf{z}$, *along with their* $k = 4$ *nearest neighbours. While* $\mathbf{x}$ *and* $\mathbf{y}$ *are not similar, their neighbours correspond to similar words. We have* $neigh(\mathbf{x}) = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4\}$ *and* $neigh(\mathbf{y}) = \{\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3, \mathbf{y}_4\}$. *We thus find:*

$$freq(diver, neigh(\mathbf{x})) = 1 \qquad freq(diver, neigh(\mathbf{y})) = 1$$
$$freq(shark, neigh(\mathbf{x})) = 1 \qquad freq(shark, neigh(\mathbf{y})) = 1$$
$$freq(submarine, neigh(\mathbf{x})) = 1 \qquad freq(submarine, neigh(\mathbf{y})) = 0$$
$$freq(coral, neigh(\mathbf{x})) = 1 \qquad freq(coral, neigh(\mathbf{y})) = 2$$

with the frequencies for all other concepts being 0. We thus obtain:

$$\pi(\mathbf{x}, \mathbf{y}) = \frac{1 + 1 + 0 + 1}{1 + 1 + 1 + 2} = \frac{3}{5}$$

As $\pi(\mathbf{x}, \mathbf{y})$ *is rather high, we will aim to move* $\mathbf{x}$ *and* $\mathbf{y}$ *closer together. In particular,* $\mathbf{x}$ *should be closer to* $\mathbf{y}$ *than to* $\mathbf{z}$, *despite the fact that* $\mathbf{x}$ *and* $\mathbf{z}$ *correspond to the same word.*

In the following, we write $Pos \subseteq (S \times V) \times (S \times V)$ to denote the resulting set of positive examples. Note that the elements of Pos are pairs of sentence-concept pairs. In particular, we have:

$$Pos = \{((s_1, c_1), (s_2, c_2)) \mid \pi(\phi(s_1, c_1), \phi(s_2, c_2)) \geq \theta, s_1 \neq s_2\}$$

for some threshold $\theta > 0$.

4.1.2 Distant Supervision from ConceptNet. We now consider a strategy which uses ConceptNet [49] as a distant supervision signal to identify positive training examples. ConceptNet contains a large number of triples of the form ([concept], *HasProperty*, [property]). We first collected all the concept-property pairs that appear in such triples. We then removed those concept-property pairs for which the property only appears for at most two concepts. Let T be the resulting set of concept-property pairs. For each pair $(c, p) \in T$, we identified all sentences in Wikipedia that mention both the concept c and the property p. We rely on the simplifying assumption that such sentences express the knowledge that concept c has property p, similar to the standard assumption underpinning distant supervision strategies for relation extraction [43]. Let S_p be the resulting set of sentence-concept pairs for property p, i.e. $(s, c) \in S_p$ if sentence s mentions both the concept c and some property p such that

$(c, p) \in T$. The set of positive examples is then defined as follows:

$$Pos = \{((s_1, c_1), (s_2, c_2)) \mid \exists p\,.\,(s_1, c_1) \in S_p, (s_2, c_2) \in S_p, s_1 \neq s_2\}$$

In other words, (s_1, c_1) and (s_2, c_2) are treated as a positive example if (i) the sentences s_1 and s_2 mention the same property p and (ii) the corresponding target concepts c_1 and c_2 have p in ConceptNet.

4.2 Fine-tuning Strategies

We now describe how the positive examples that were identified in Section 4.1 can be used for fine-tuning the mention vectors. The most straightforward strategy, which we discuss in Section 4.2.2, is based on fine-tuning the language model itself. The main drawback of this method is that it is computationally expensive. For this reason, in Section 4.2.1 we first discuss a simpler strategy, which simply learns a linear projection of the standard mention vectors.

4.2.1 Projection Method. Our aim is to learn a projection matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ such that vectors $\mathbf{A}\phi(s_1, c_1)$ and $\mathbf{A}\phi(s_2, c_2)$ are similar iff $((s_1, c_1), (s_2, c_2)) \in Pos$. Here n is the dimension of mention vectors while m is the dimension of the resulting vectors. We can think of $\mathbf{A}$ as selecting the subspace of the mention vector space that is focused on semantic properties. We use the supervised contrastive loss from Khosla et al. [26] to learn $\mathbf{A}$. Let $B \subseteq S \times V$ be the set of sentence-concept pairs that are considered in a given mini-batch. Let $X_{(s,c)} = \{(s', c') \mid ((s, c), (s', c')) \in Pos \cap (B \times B)\}$ be the set of positive examples for (s, c) in the mini-batch. The loss is as follows:

$$\sum_{(s,c)\in B} \frac{-1}{|X_{(s,c)}|} \sum_{(s',c')\in X_{(s,c)}} \log \frac{e^{\cos(\mathbf{A}\phi(s,c),\mathbf{A}\phi(s',c'))/\tau}}{\sum_{(s'',c'')} e^{\cos(\mathbf{A}\phi(s,c),\mathbf{A}\phi(s'',c''))/\tau}}$$

where the summation in the denominator ranges over $(s'', c'') \in B \setminus \{(s, c)\}$, and the temperature $\tau > 0$ is a hyperparameter.

4.2.2 Fine-Tuning BERT. We now consider a variant in which the contrastive loss is used to fine-tune a BERT encoder. This should allow us to learn more informative mention vectors, but at a higher computational cost. Let us write $\psi(s, c)$ for the encoding of sentence-concept pair (s, c) according to the fine-tuned BERT encoder (to distinguish it from ϕ, which uses the pre-trained language model). Let B and $X_{(s,c)}$ be defined as before. We use the following loss:

$$\sum_{(s,c)\in B} \frac{-1}{|X_{(s,c)}|} \sum_{(s',c')\in X_{(s,c)}} \log \frac{e^{\cos(\psi(s,c),\psi(s',c'))/\tau}}{\sum_{(s'',c'')} e^{\cos(\psi(s,c),\psi(s'',c''))/\tau}}$$

where the summation in the denominator ranges over $(s'', c'') \in B \setminus \{(s, c)\}$, as before, and $\tau > 0$ is again a hyperparameter.

5 EXPERIMENTS

We present an evaluation of our proposed strategies[2]. We will in particular focus on the following variants:

- **ConProj** uses the projection method for fine-tuning and the neighbourhood structure for obtaining positive examples.
- **ConFT** fine-tunes the BERT encoder and uses the neighbourhood structure for obtaining positive examples.
- **ConCN** fine-tunes the BERT encoder and uses the distant supervision strategy based on Conceptnet for obtaining positive examples.

[2]Datasets and code at https://github.com/lina-luck/semantic_concept_embeddings.

By comparing these variants we are particularly interested in answering the following two research questions: (i) is learning a linear projection sufficient or do we need to fine-tune the language model, and (ii) how effective are the two proposed strategies for obtaining weakly labelled positive examples. The primary focus of our experiments is on word classification (Section 5.1), as these allow us to directly evaluate the extent to which our embeddings capture different kinds of semantic properties. This is motivated by the observation that this is precisely what matters in most applications where static concept embeddings are still needed. For instance, tasks such as ontology completion or zero-shot learning directly use concept embeddings to link concepts to their semantic properties. We also evaluate the quality of the clusters that arise from our embeddings (Section 5.2). To verify that the concept embeddings are indeed useful in downstream applications, we present an evaluation on the downstream task of ontology completion (Section 5.3). We conclude with an analysis of the main results (Section 5.4).

Baselines. We compare our embeddings with Skip-gram [42] and GloVe [47], as representative examples of traditional word embeddings[3], and with SynGCN[4] [52] and Word2Sense[5] [46], as examples of more recent static word embeddings. We furthermore compare with the Numberbatch[6] embeddings from Speer et al. [49], as these were also fine-tuned based on ConceptNet. Beyond traditional word embeddings, we compare with the method from Li et al. [31], as we use their mention vectors as our starting point. We include two variants: one version where all mention vectors are averaged (*Mask*) and one version where their filtering strategy is applied first (*Mask+filtering*). In addition, we consider a variant in which mention vectors are obtained without masking the target concept (*No-Mask*). In this case, for words that consist of more than one token, the contextualised token representations are averaged. Rather than taking the final layer representation, which has been found to be sub-optimal [55], in this case, we select the optimal layer based on a validation split. Finally, we include results for MirrorBERT[7] [33] and MirrorWiC[8] [34], both of which also use a contrastively fine-tuned BERT model.

Training Details. To obtain mention vectors, for each concept, we randomly sample up to 500 sentences mentioning that concept from Wikipedia. We use the same sentences for our methods, for the baseline methods from Li et al. [31] and for MirrorWiC. Unless specified otherwise, we use BERT-large-uncased as the pre-trained language model. The learning rate for our models was set to 2e-4, with cosine warm-up for the first 2 epochs. We use early stopping with a patience of 10 and a minimum difference of 1e-10. We used the AdamW optimizer. We set the temperature parameter in the contrastive loss to 0.05 and the number of neighbours k for evaluating the compatibility degree to 5. The threshold θ on compatibility degrees to be considered a positive example was set to 0.5. For

[3]We used the Skip-gram embeddings trained on Google News (https://code.google.com/archive/p/word2vec/) and Glove embeddings trained on Common Crawl (https://nlp.stanford.edu/projects/glove/).
[4]https://drive.google.com/file/d/1wYgdyjIBC6nIC-bX29kByA0GwnUSR9Hh/view
[5]https://drive.google.com/file/d/1kqxQm129RVfanlnEsJnyYjygsFhA3wH3/view
[6]https://conceptnet.s3.amazonaws.com/downloads/2019/numberbatch/numberbatch-en-19.08.txt.gz
[7]https://huggingface.co/cambridgeltl/mirror-bert-base-uncased-word
[8]https://huggingface.co/cambridgeltl/mirrorwic-bert-base-uncased

Table 1: Results (%) for BERT-large-uncased on the lexical classification tasks, in terms of F1 (%).

	X-McRae		CSLB		Morrow		WNSS		BabelDom		BM		AP	
	SVM	CNN	SVM	CNN	SVM	CNN	SVM	CNN	SVM	CNN	SVM	CNN	SVM	CNN
GloVe	63.6	-	42.7	-	57.1	-	48.6	-	41.9	-	59.4	-	60.7	-
Skip-Gram	61.3	-	50.2	-	64.7	-	55.9	-	49.3	-	60.3	-	61.7	-
Word2Sense	52.3	-	50.3	-	69.2	-	43.9	-	32	-	63.8	-	62.1	-
SynGCN	56.5	-	50.9	-	71.4	-	42.3	-	34.2	-	76.2	-	75.6	-
Numberbatch	63.5	-	57.8	-	71.1	-	63.4	-	41.5	-	80.7	-	82.3	
MirrorBERT	63.3	-	51.6	-	69.8	-	59.1	-	50.3	-	79.2	-	82.8	-
MirrorWiC	64.2	67.6	52.7	60.1	70.6	79.3	59.1	63.4	50.4	56.2	80.1	81.6	81.4	82.6
No-Mask	55.9	57.3	45.6	46.8	67.5	68.2	50.9	51.8	40.3	42.4	67.2	68.4	62.5	64.1
Mask	62.8	66.8	44.8	47.2	57.8	59.3	56.5	57.3	49.3	51.1	78.6	80.1	79.3	81.9
Mask + filtering	64.1	67.7	51.4	54.3	73.5	75.4	58.5	61.3	50.9	53.6	79.6	82.6	81.9	82.3
ConProj	66.6	69.3	53.6	61.4	75.5	81.1	63.2	65.8	54.7	58.4	80.6	82.7	82.9	83.8
ConFT	67.4	69.8	55.7	63.6	76.9	82.4	65.7	67.2	55.8	59.6	81.1	82.9	83.3	84.2
ConCN	68.3	70.9	56.2	65.1	77.5	83.8	67.1	69.4	57.3	61.7	81.8	83.6	84.1	85.3
ConProj + filt.	70.1	73.2	56.3	68.8	78.8	83.7	65.2	68.6	59.1	63.9	81.2	83.3	83.4	84.6
ConFT + filt.	71.9	74.4	57.3	69.3	78.5	86.2	67.1	69.3	60.7	64.8	82.1	83.8	84.1	85.1
ConCN + filt.	**73.7**	**75.2**	**59.4**	**71.8**	**81.1**	**87.5**	**68.9**	**70.8**	**62.5**	**67.1**	**83.2**	**84.7**	**84.7**	**85.9**

implementing the contrastive loss, we relied on the Pytorch Metric Learning library[9]. Based on the values reported by Li et al. [31], we set the number of neighbours for the filtering strategy to 50 for X-McRae, WNSS and BabelDomains, and 5 for CSLB, Morrow, BM and AP. The dimension m of the transformed vectors, for the projection-based fine-tuning method, is 256. For *ConProj*, we obtain the sentence-concept pairs for a given mini-batch by sampling 1024 such pairs from the set *Pos*. For *ConFT*, we proceed similarly, but limit the number of pairs to 512 due to memory constraints. For *ConCN*, the set of sentence-concept pairs for a given mini-batch is obtained by repeatedly (i) sampling a property p and (ii) sampling 50 sentences from $S(p)$.

5.1 Word Classification

We consider a number of benchmarks which involve predicting whether a given concept belongs to some class, where the classes of interest correspond to different kinds of semantic properties, namely commonsense properties (e.g. being made of wood), taxonomic categories (e.g. being an animal) and thematic domains (e.g. related to music). We evaluate the extent to which these classes can be predicted from different kinds of concept embeddings. We have included the five benchmarks that were used by Li et al. [31]:

- the extension of the McRae feature norms [40] that was introduced by Forbes et al. [14] (X-McRae[10]), covering 513 words and 50 classes (being commonsense properties);
- CSLB Concept Property Norms[11], with 635 words and 395 classes (being commonsense properties);
- the Morrow dataset [45], covering 888 words and 13 classes (being broad taxonomic categories such as *animals*);
- WordNet supersenses[12] (WNSS), with 18200 words and 25 classes (being broad taxonomic categories);
- BabelDomains[13] [7], covering 12477 words and 28 classes (being thematic domains).

For these datasets, we use the same training-tuning-test splits as Li et al. [31][14]. We also include two additional benchmarks[15]:

- the Battig and Montague norms [4], with 5321 words and 56 classes (being fine-grained taxonomic categories such as *weapon* or *unit of time*);
- the dataset from Almuhareb and Poesio [2], with 402 words and 21 classes (being WordNet hypernyms).

For both datasets, we randomly split the positive examples, for each category, into 60% for training, 20% for tuning and 20% for testing. As these datasets only specify positive examples, for each concept, we generate 5 negative examples by randomly selecting categories to which the concept does not belong to.

Methodology. For each class, we train a linear SVM to classify concepts based on their embedding. We report the results in terms of F1 score, macro-averaged across all classes from a given benchmark. We furthermore experiment with a simple Convolutional Neural Network (CNN), which takes the individual mention vectors as input, rather than their average. In particular, each mention vector is first fed through a dense layer and the resulting vectors are aggregated using max-pooling. This aggregated vector is then fed to a classification layer. For the SVM, we used the standard scikit-learn implementation. The C parameter is tuned from $\{0.1, 1, 10, 100\}$. For the CNN model, we have used the standard PyTorch implementation, setting the kernel size and stride to 1. We used 64 filters with ReLU activation, a batch size of 32 and a learning rate of 1e-3. The CNN is trained with binary cross-entropy, using Adam.

[9] https://kevinmusgrave.github.io/pytorch-metric-learning/
[10] https://github.com/mbforbes/physical-commonsense
[11] https://cslb.psychol.cam.ac.uk/propnorms
[12] https://wordnet.princeton.edu/download
[13] http://lcl.uniroma1.it/babeldomains/
[14] It should be noted that the annotations in CSLB are not complete, i.e. some properties which are not asserted to hold for a given concept are nonetheless valid [44]. This means that care is needed when drawing conclusions from the absolute performance of models on this dataset. As we are mostly interested in the relative performance of different embeddings in this paper, this should not affect the analysis.
[15] We used the versions available at https://github.com/vecto-ai/word-benchmarks.

Results. The results are summarised in Table 1. A number of clear observations can be made. First, all three of the proposed methods (*ConProj*, *ConFT*, *ConCN*) outperform the baselines[16]. The main exception is CSLB, where Numerbatch outperforms all SVM-based models apart from *ConCN* with filtering. Among our proposed methods, *ConCN* performs best, showing the effectiveness of the ConceptNet-based distant supervision strategy, while *ConFT* outperforms *ConProj*, as expected. As a second observation, the filtering strategy from Li et al. [31] is highly effective, offering improvements that are complementary to those of our proposed methods. Third, the CNN consistently outperforms the SVM model, with the margin being particularly large for CSLB.

5.2 Clustering

The BM [4] and AP [2] datasets, which we used for word classification, have also been used as clustering benchmarks in previous work [3]. Specifically, the aim is to organise the words from the dataset into semantically meaningful clusters. The clusters are evaluated using cluster purity, using the categories which are provided in the dataset as the ground truth. The main aim of this experiment is to analyse the quality of our embeddings in an unsupervised setting, to test their suitability for tasks such as topic modelling [9, 11, 63]. We use k-means to obtain the clusters, choosing k as the number of categories from the dataset. Since the quality of the clusters is sensitive to the random initialisation of the clusters, we repeat the experiment 10 times and report the average purity.

The results are shown in Table 2. As can be seen, our method outperforms all baselines. Similar as for word classification, we can see that ConCN is the best variant and that the filtering strategy consistently improves the results. Among the baselines, the strong performance of Numberbatch is also notable.

5.3 Ontology Completion

An ontology can be viewed as a set of rules. A simple rule takes the following form:

$$A_1(x) \wedge ...A_n(x) \rightarrow B(x)$$

It expresses the knowledge that whenever some entity x belongs to the concepts $A_1, ..., A_n$ then it also belongs to the concept B. In general, rules may also contain constructs of the form $\exists y\ R(x, y) \wedge A(y)$, which expresses that x is related, via relation R, to some instance of A. The key principle underpinning the ontology completion benchmarks from [32] is that real-world ontologies often contain sets of closely related rules, which only differ in a single concept. Consider, for instance, an ontology containing the following rules:

$$\mathit{AppleJuice}(x) \wedge \mathit{Small}(x) \rightarrow \mathit{SuitableForKids}(x)$$
$$\mathit{PineappleJuice}(x) \wedge \mathit{Small}(x) \rightarrow \mathit{SuitableForKids}(x)$$
$$\mathit{MangoJuice}(x) \wedge \mathit{Small}(x) \rightarrow \mathit{SuitableForKids}(x)$$

For instance, the first rule intuitively captures the knowledge that a small portion of apple juice is suitable for kids to drink. From these rules, we may infer that the following rule should also be considered valid within the context of this ontology, even if it is not actually provided:

$$\mathit{OrangeJuice}(x) \wedge \mathit{Small}(x) \rightarrow \mathit{SuitableForKids}(x)$$

The underlying principle is that *orange juice* satisfies all the properties that are common to *apple juice*, *pineapple juice* and *mango juice*. To infer such plausible rules, we often need to combine prior knowledge about the meaning of the concepts (e.g. that orange juice has similar properties to apple juice and pineapple juice) with the knowledge that is inferred from the structure of the ontology itself (e.g. to deal with concepts whose name is not descriptive). To this end, [32] introduced a graph neural network, in which the nodes correspond to concepts. Concepts that co-occur in the same rule are connected with an edge. The input representation of each node is a pre-trained concept embedding, which was taken to be the skip-gram embedding of the concept name in [32]. Ontology completion has a number of practical applications. For instance, apart from suggesting plausible missing knowledge to ontology engineers, the ability to predict plausible rules also plays an important role in ontology alignment [21].

Following, [31], we use ontology completion benchmarks for evaluating the quality of different types of concept embeddings, using the same methodology. In particular, we first tokenise concept names, based on the common naming conventions in ontologies. For instance, the concept *PastaWithWhiteSauce* becomes "pasta with white sauce". If the resulting concept name does not appear in Wikipedia, we never predict this concept as a positive example. We use the same hyperparameters and training-test splits as [32], and use their evaluation scripts[17]. The benchmark includes five different ontologies. First, the SUMO ontology was included as a prototypical example of a large open-domain ontology. The other

Table 2: Results for clustering and ontology completion using BERT-large-uncased. Clustering results are in terms of purity (%) while ontology completion results are in terms of F1 (%).

	Clustering		Ontology Completion				
	BM	**AP**	**Wine**	**Econ**	**Olym**	**Tran**	**SUMO**
Glove	57.3	44.9	14.2	14.1	9.9	8.3	34.9
Skip-Gram	46.7	32.4	13.8	13.5	8.3	7.2	33.4
Word2Sense	25.5	16.6	13.4	13.2	8.1	7.2	33.1
SynGCN	56.9	39.2	13.9	13.8	9.4	8.1	33.9
Numberbatch	73.8	53.3	25.6	26.2	26.8	16.0	47.3
MirrorBERT	62.4	51.4	22.5	23.8	20.9	12.7	40.1
MirrorWiC	64.6	52.5	24.7	24.9	22.1	13.9	46.9
Mask + filt.	61.3	48.2	24.5	24.3	22.9	13.0	46.4
ConProj	75.8	54.2	26.9	27.3	25.6	15.9	48.2
ConFT	76.1	56.9	27.5	29.2	26.5	17.4	48.6
ConCN	76.9	57.2	29.1	31.3	27.6	19.7	50.4
ConProj + filt.	76.3	54.9	27.2	28.6	26.2	17.1	49.3
ConFT + filt.	76.8	57.3	28.7	30.3	28.2	19.1	50.3
ConCN + filt.	**77.4**	**57.9**	**31.3**	**32.4**	**29.7**	**20.9**	**52.6**

[16]Note that *MirrorBERT* and *MirrorWiC* use BERT-base, whereas our models and those from Li et al. [31] rely on BERT-large. However, as we will see below, the outperformance of our model remains after changing the encoder to BERT-base. We use BERT-large for the main experiments, as the methods from Li et al. [31], which are our primary baselines, achieve substantially weaker results for BERT-base.

[17]https://github.com/lina-luck/rosv_ijcai21

Table 3: Comparison of different language models and strategies for selecting positive examples, for X-McRae, in terms of F1 (%). Results are for BERT-base-uncased (BB), BERT-large-uncased (BL), RoBERTa-base (RB) and RoBERTa-large (RL).

	BB		BL		RB		RL	
	SVM	CNN	SVM	CNN	SVM	CNN	SVM	CNN
No-Mask	49.6	51.2	55.9	57.3	50.4	52.4	53.5	57.2
Mask	53.6	57.2	62.8	66.8	52.1	54.6	63.9	67.1
Mask + filtering	58.2	60.3	64.1	67.7	59.5	61.8	64.8	68.1
ConProj	64.3	67.6	66.6	69.3	64.9	67.9	67.2	69.8
ConFT	65.2	68.1	67.4	69.8	65.3	68.2	67.4	70.1
ConCN	66.4	69.5	68.3	70.9	67.2	70.0	69.6	71.3
ConProj + filt.	66.3	68.3	70.1	73.2	67.7	69.4	70.5	73.9
ConFT + filt.	67.0	70.1	71.9	74.4	68.2	71.6	71.3	73.5
ConCN + filt.	**68.3**	**72.5**	**73.7**	**75.2**	**69.1**	**73.3**	**73.9**	**75.8**
W-ConProj	61.2	65.9	64.9	68.7	62.1	66.2	65.9	69.2
W-ConProj + filt.	63.8	67.3	68.6	71.9	64.5	69.2	70.1	73.6

four are well-known domain-specific ontologies: Wine, Economy, Olympics and Transport[18].

The results for ontology completion in Table 2 are broadly in line with those from the word classification and clustering experiments. Note in particular how the performance of *ConCN + filt*, our best-performing variant, is substantially higher than that of *Numberbatch*, *MirrorBERT*, *MirrorWiC* and *Mask + filt*, which in turn substantially outperform the remaining baselines. Overall, these results clearly show that high-quality concept embeddings can be extracted from language models, which have significant benefits over traditional word embeddings. For instance, with the exception of SUMO, all our methods achieve F1 scores which at least double the F1 scores of skip-gram. Moreover, compared to earlier BERT-based methods such as *MirrorBERT*, *MirrorWiC* and *Mask*, our vectors are more focused on the semantic properties of concepts, which gives them a clear advantage in this task.

5.4 Analysis

We now present some additional analysis of our models, focusing primarily on the results for word classification.

Outperformance of the CNN. The CNN is expected to outperform when the semantic properties we need to predict are only rarely mentioned in text. Indeed, such properties will only be captured by a small number of mention vectors, and this information will be largely lost after averaging them. CSLB focuses on commonsense properties, many of which are indeed rarely expressed in text [18], which explains the large outperformance of the CNN model for this benchmark (as well as the comparatively strong performance of Numberbatch) in Table 1. For instance, the categories for which the difference in F1 score between the SVM and CNN models is largest, for *ConCN+filtering*, are as follows: *grows on plants*, *is cool*, *has a top*, *is furry*, *has green leaves*, *is for soup*, *is ridden*, *is a body part*, *is found in America*, *has big eyes*, *has arms*, *has a blade/blades*. For X-McRae, the overall differences are smaller, which can be explained by the fact that several taxonomic properties are included in this dataset as well. However, for many commonsense properties, we similarly observe large differences in F1 score. The largest differences were observed for the following X-McRae properties: *loud*, *used for holding things*, *words on it*, *eaten in summer*, *worn for warmth*, *flies*, *used for killing*, *used for cleaning*, *worn on feet*.

Comparing Language Models. Table 3 analyses the impact of changing the language model encoder, showing word classification results for BERT-base-uncased, BERT-large-uncased, RoBERTa-base and RoBERTa-large [36], for the SVM model. We can see that BERT-large and RoBERTa-large outperform the base models, as expected, but the differences for our methods are relatively small. In contrast, for the *No-Mask*, *Mask* and *Mask+filtering* baselines, switching to the base models is more detrimental. Across all language models, we find that our proposed methods outperform the baselines.

Importance of the Compatibility Degree. For the *ConProj* and *ConFT* variants, the set of positive examples is based on the neighbourhood structure of the mention vectors (see Section 4.1.1). Another possibility could be to simply assume that sentences mentioning the same word are likely to express the same property. In other words, we could define the set of positive examples as follows:

$$Pos = \{((s_1, c), (s_2, c)) \mid s_1, s_2 \in S, c \in V, s_1 \neq s_2\}$$

The effectiveness of this alternative strategy is analysed in Table 3, where it is referred to as *W-ConProj* (when used in combination with the projection-based contrastive loss). While this alternative strategy also outperforms the baselines, it consistently underperforms our main neighbourhood-based strategy.

Anisotropy. As mentioned in the introduction, one of the reasons for the underperformance of the *Mask* embeddings may be related to the high anisotropy of the BERT mention vectors. Figure 2 shows a histogram of the cosine similarities between randomly sampled concept embeddings, for the *Mask* and *ConCN* strategies. As can be seen, the cosine similarities are on average lower for *ConCN*, which shows that this contrastive learning strategy has indeed led to a reduction in anisotropy.

Qualitative Analysis. We now explore how the mention vectors are affected by the proposed fine-tuning strategy. Specifically, we consider pairs (s_1, c_1), (s_2, c_2) where the mention vector for (s_2, c_2) is among the top-100 nearest neighbours of the mention vector for (s_1, c_1) when using *ConCN*, while not being among the top-1000 nearest neighbours when using *Mask* (for the full set of mention vectors M across all words). Table 4 contains some examples of such sentence pairs. The examples illustrate how fine-tuning allows the model to identify sentences that express similar properties, even when the sentences themselves are not similar, neither in syntactic structure nor in their overall meaning. In the first example, both sentences express that the target concept (which is masked) is some kind of building. Similarly, in the second example, the sentences express that the target concepts can be black. The third example illustrates a more abstract property, capturing the fact that country-specific versions of the target concept exist.

[18]We used the training and test splits from https://github.com/bzdt/GCN-based-Ontology-Completion.

Table 4: The table shows pairs of sentences whose mention vectors are similar when using the model fine-tuned with the ConCN strategy while being dissimilar when using pre-trained BERT model.

Similar after fine-tuning but not for pre-trained model	
The second floor of the facade was originally designed to be a private Disney family apartment.	It would also allow the RTC to buy new curtains and wall coverings, and to restore the building's exterior.
A vinaigrette can be made with black garlic, sherry vinegar, soy, a neutral oil, and Dijon mustard.	... in flood situations where normal foods are out of reach black swans will feed on pasture plants on shore.
... raising the value of the Icelandic crown in 1925, very much as Winston Churchill raised the value of the pound ...	There are further plans to reintroduce the South African cheetahs to the Lower Zambezi.

Table 5: Nearest neighbours, in terms of cosine similarity, for some selected words, using WordNet supersenses vocabulary.

	Word	Neighbours
Numberbatch	lemon	citron, citrange, limeade, lime, lemonade
	deepening	broadening, deep, strengthening, deepness, worsening
	icon	iconology, symbol, iconography, iconoclasm, emblem
	stunt	trick, aerialist, jugglery, gimmickry, cartwheel
	milkman	dairyman, milk, creamery, clabber, lacteal
	paradox	antinomy, contradiction, duality, oxymoron, inconsistency
	desk	office, copyholder, desktop, bookcase, table
	beer	ale, brewery, microbrewery, brewpub, keg
	steam	steamer, steamboat, steamfitter, gasification, boiling
	razor	razorblade, shaver, blade, scissors, sharpener
MirrorBERT	lemon	lemonwood, lemonade, orangeade, limeade, dewberry
	deepening	deepness, broadening, deep, shallowness, diversification
	icon	iconoclast, iconography, iconoclasm, iconology, symbol
	stunt	trick, props, joyride, sabotage, leap
	milkman	dairyman, milk, alewife, grocer, milkwort
	paradox	contradiction, ambiguity, perplexity, singularity, unreality
	desk	office, clerk, bookcase, counter, receptionist
	beer	ale, liquor, rum, brewpub, brandy
	steam	steamfitter, boilerplate, turbine, generator, gasification
	razor	razorblade, blade, scissors, needle, knife
ConCN + filt.	lemon	lime, blueberry, tangerine, cranberry, lemonade
	deepening	broadening, weakening, mellowing, narrowing, depths
	icon	button, plaque, emblem, display, iconography
	stunt	handstand, gimmickry, skydiver, fling, skydiving
	milkman	cheesemonger, dairyman, barmaid, paperboy, cow
	paradox	singularity, irony, doublethink, unreality, perplexity
	desk	counter, sideboard, office, bookcase, drawer
	beer	mead, ale, vodka, brandy, tequila
	steam	electricity, furnace, turbine, vent, gasification
	razor	penknife, tool, scalpel, razorblade, shaver

Finally, Table 5 shows the nearest neighbours of some selected target words, in terms of cosine similarity, for three different concept embeddings: *Numberbatch*, *MirrorBERT* and *ConCN* (with filtering). For this analysis, we considered the vocabulary from the WordNet supersenses dataset. A first observation is that the neighbours for *ConCN* are often taxonomically closer. For instance, for *MirrorBERT* we see *lemonwood* as a top neighbour of *lemon*, which is topically related but not taxonomically close. Similarly, for both *Numberbatch* and *MirrorBERT* we see *milk* as the second nearest neighbour of *milkman*. As another difference, for *ConCN* we can see neighbours which involve some abstraction. For instance, a *button* has a similar role as an *icon* in graphical user interfaces. Another notable example is *cow* as a neighbour of *milkman*, which are both related to the production/delivery of milk. However, this notion of abstraction sometimes also leads to sub-optimal neighbours. For instance, *contradiction* is shown as one of the top neighbours of *paradox* for both *Numberbatch* and *MirrorBERT* but does not appear as a neighbour for *ConCN*.

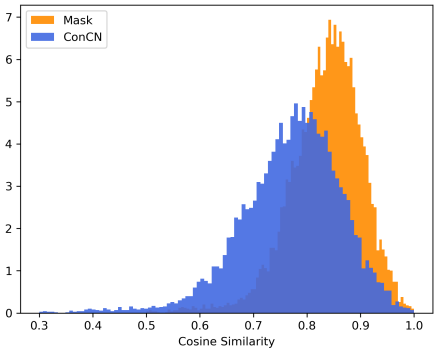

Figure 2: Histogram of cosine similarities between the embeddings of two randomly sampled concepts, chosen from those appearing in the X-McRae, for the *Mask* and *ConCN*.

6 CONCLUSIONS

We have proposed a method for learning concept embeddings, based on contextualised representations of masked mentions of concepts in a text corpus. Our focus was on improving the contextualised representations that can be obtained from a pre-trained BERT model, using a number of strategies based on contrastive learning. The aim of these strategies is to ensure that two contextualised word embeddings are similar if and only if the corresponding sentences express similar properties. To implement this idea, we need examples of sentences that are likely to express the same property. We have proposed two methods for obtaining such examples: an unsupervised method which relies on the neighbourhood structure of contextualised word vectors, and a distantly supervised method which relies on ConceptNet. In our experimental results, we found the latter method to perform best. Our proposed strategy was also found to outperform a range of baselines, both in word classification experiments and for the task of ontology completion.

ACKNOWLEDGMENTS

This work was supported by ANR-22-CE23-0002 ERIANA and EPSRC grant EP/V025961/1. Na Li is supported by Shanghai Big Data Management System Engineering Research Center Open Funding.

REFERENCES

[1] Israa Alghanmi, Luis Espinosa Anke, and Steven Schockaert. 2020. Combining BERT with Static Word Embeddings for Categorizing Social Media. In *Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020)*. Association for Computational Linguistics, Online, 28–33. https://doi.org/10.18653/v1/2020.wnut-1.5

[2] Abdulrahman Almuhareb and Massimo Poesio. 2005. Concept learning and categorization from the web. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, Vol. 27.

[3] Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 2014. Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Baltimore, Maryland, 238–247. https://doi.org/10.3115/v1/P14-1023

[4] William F Battig and William E Montague. 1969. Category norms of verbal items in 56 categories: A replication and extension of the Connecticut category norms. *Journal of Experimental Psychology* 80 (1969), 1–46.

[5] Sumithra Bhakthavatsalam, Chloe Anastasiades, and Peter Clark. 2020. GenericsKB: A Knowledge Base of Generic Statements. *CoRR* abs/2005.00660 (2020).

[6] Rishi Bommasani, Kelly Davis, and Claire Cardie. 2020. Interpreting Pretrained Contextualized Representations via Reductions to Static Embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 4758–4781. https://doi.org/10.18653/v1/2020.acl-main.431

[7] Jose Camacho-Collados and Roberto Navigli. 2017. BabelDomains: Large-Scale Domain Labeling of Lexical Resources. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*. Association for Computational Linguistics, Valencia, Spain, 223–228. https://aclanthology.org/E17-2036

[8] Jiaoyan Chen, Yuan He, Ernesto Jiménez-Ruiz, Hang Dong, and Ian Horrocks. 2022. Contextual Semantic Embeddings for Ontology Subsumption Prediction. *CoRR* abs/2202.09791 (2022). arXiv:2202.09791 https://arxiv.org/abs/2202.09791

[9] Rajarshi Das, Manzil Zaheer, and Chris Dyer. 2015. Gaussian LDA for Topic Models with Word Embeddings. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, Beijing, China, 795–804. https://doi.org/10.3115/v1/P15-1077

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. https://doi.org/10.18653/v1/N19-1423

[11] Adji Bousso Dieng, Francisco J. R. Ruiz, and David M. Blei. 2020. Topic Modeling in Embedding Spaces. *Trans. Assoc. Comput. Linguistics* 8 (2020), 439–453. https://doi.org/10.1162/tacl_a_00325

[12] Kawin Ethayarajh. 2019. How Contextual are Contextualized Word Representations? Comparing the Geometry of BERT, ELMo, and GPT-2 Embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, Hong Kong, China, 55–65. https://doi.org/10.18653/v1/D19-1006

[13] John R Firth. 1957. A synopsis of linguistic theory, 1930-1955. *Studies in Linguistic Analysis* (1957).

[14] Maxwell Forbes, Ari Holtzman, and Yejin Choi. 2019. Do Neural Language Representations Learn Physical Commonsense?. In *Proceedings of the 41th Annual Meeting of the Cognitive Science Society*, Ashok K. Goel, Colleen M. Seifert, and Christian Freksa (Eds.). cognitivesciencesociety.org, 1753–1759. https://mindmodeling.org/cogsci2019/papers/0311/index.html

[15] Amit Gajbhiye, Luis Espinosa-Anke, and Steven Schockaert. 2022. Modelling Commonsense Properties Using Pre-Trained Bi-Encoders. In *Proceedings of the 29th International Conference on Computational Linguistics*. International Committee on Computational Linguistics, Gyeongju, Republic of Korea, 3971–3983. https://aclanthology.org/2022.coling-1.349

[16] Leilei Gan, Zhiyang Teng, Yue Zhang, Linchao Zhu, Fei Wu, and Yi Yang. 2020. SemGloVe: Semantic Co-occurrences for GloVe from BERT. *CoRR* abs/2012.15197 (2020). arXiv:2012.15197 https://arxiv.org/abs/2012.15197

[17] Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple Contrastive Learning of Sentence Embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 6894–6910. https://doi.org/10.18653/v1/2021.emnlp-main.552

[18] Jonathan Gordon and Benjamin Van Durme. 2013. Reporting bias and knowledge acquisition. In *AKBC@CIKM*. ACM, 25–30.

[19] Prakhar Gupta and Martin Jaggi. 2021. Obtaining Better Static Word Embeddings Using Contextual Embedding Models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, Online, 5241–5253. https://doi.org/10.18653/v1/2021.acl-long.408

[20] Zellig S Harris. 1954. Distributional structure. *Word* 10, 2-3 (1954), 146–162.

[21] Yuan He, Jiaoyan Chen, Denvar Antonyrajah, and Ian Horrocks. 2022. BERTMap: A BERT-Based Ontology Alignment System. 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*. AAAI Press, 5684–5691. https://ojs.aaai.org/index.php/AAAI/article/view/20510

[22] John Hewitt and Christopher D. Manning. 2019. A Structural Probe for Finding Syntax in Word Representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, 4129–4138. https://doi.org/10.18653/v1/N19-1419

[23] Yutai Hou, Wanxiang Che, Yongkui Lai, Zhihan Zhou, Yijia Liu, Han Liu, and Ting Liu. 2020. Few-shot Slot Tagging with Collapsed Dependency Transfer and Label-enhanced Task-adaptive Projection Network. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 1381–1393. https://doi.org/10.18653/v1/2020.acl-main.128

[24] Junjie Huang, Duyu Tang, Wanjun Zhong, Shuai Lu, Linjun Shou, Ming Gong, Daxin Jiang, and Nan Duan. 2021. WhiteningBERT: An Easy Unsupervised Sentence Embedding Approach. In *Findings of the Association for Computational Linguistics: EMNLP 2021*. Association for Computational Linguistics, Punta Cana, Dominican Republic, 238–244. https://doi.org/10.18653/v1/2021.findings-emnlp.23

[25] Lei Ji, Yujing Wang, Botian Shi, Dawei Zhang, Zhongyuan Wang, and Jun Yan. 2019. Microsoft Concept Graph: Mining Semantic Concepts for Short Text Understanding. *Data Intell.* 1, 3 (2019), 238–270. https://doi.org/10.1162/dint_a_00013

[26] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised Contrastive Learning. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (Eds.). https://proceedings.neurips.cc/paper/2020/hash/d89a66c7c80a29b1bdbab0f2a1a94af8-Abstract.html

[27] Prodromos Kolyvakis, Alexandros Kalousis, and Dimitris Kiritsis. 2018. DeepAlignment: Unsupervised Ontology Matching with Refined Word Vectors. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics, New Orleans, Louisiana, 787–798. https://doi.org/10.18653/v1/N18-1072

[28] Andrey Kutuzov, Lilja Øvrelid, Terrence Szymanski, and Erik Velldal. 2018. Diachronic word embeddings and semantic shifts: a survey. In *Proceedings of the 27th International Conference on Computational Linguistics*. Association for Computational Linguistics, Santa Fe, New Mexico, USA, 1384–1397. https://aclanthology.org/C18-1117

[29] Aoxue Li, Weiran Huang, Xu Lan, Jiashi Feng, Zhenguo Li, and Liwei Wang. 2020. Boosting Few-Shot Learning With Adaptive Margin Loss. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 12573–12581.

[30] Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020. On the Sentence Embeddings from Pre-trained Language Models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Online, 9119–9130. https://doi.org/10.18653/v1/2020.emnlp-main.733

[31] Na Li, Zied Bouraoui, José Camacho-Collados, Luis Espinosa Anke, Qing Gu, and Steven Schockaert. 2021. Modelling General Properties of Nouns by Selectively Averaging Contextualised Embeddings. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*, Zhi-Hua Zhou (Ed.). ijcai.org, 3850–3856. https://doi.org/10.24963/ijcai.2021/530

[32] Na Li, Zied Bouraoui, and Steven Schockaert. 2019. Ontology Completion Using Graph Convolutional Networks. In *The Semantic Web - ISWC 2019 - 18th International Semantic Web Conference, Auckland, New Zealand, October 26-30, 2019, Proceedings, Part I (Lecture Notes in Computer Science, Vol. 11778)*, Chiara Ghidini, Olaf Hartig, Maria Maleshkova, Vojtech Svátek, Isabel F. Cruz, Aidan Hogan, Jie Song, Maxime Lefrançois, and Fabien Gandon (Eds.). Springer, 435–452. https://doi.org/10.1007/978-3-030-30793-6_25

[33] Fangyu Liu, Ivan Vulić, Anna Korhonen, and Nigel Collier. 2021. Fast, Effective, and Self-Supervised: Transforming Masked Language Models into Universal Lexical and Sentence Encoders. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 1442–1459. https://doi.org/10.18653/v1/2021.emnlp-main.109

[34] Qianchu Liu, Fangyu Liu, Nigel Collier, Anna Korhonen, and Ivan Vulić. 2021. MirrorWiC: On Eliciting Word-in-Context Representations from Pretrained Language Models. In *Proceedings of the 25th Conference on Computational Natural Language Learning*. Association for Computational Linguistics, Online, 562–574. https://doi.org/10.18653/v1/2021.conll-1.44

[35] Qianchu Liu, Diana McCarthy, and Anna Korhonen. 2020. Towards Better Context-aware Lexical Semantics:Adjusting Contextualized Representations through Static Anchors. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Online, 4066–4075. https://doi.org/10.18653/v1/2020.emnlp-main.333

[36] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR* abs/1907.11692 (2019).

[37] Ziyang Luo, Artur Kulmizev, and Xiaoxi Mao. 2021. Positional Artefacts Propagate Through Masked Language Model Embeddings. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, Online, 5312–5327. https://doi.org/10.18653/v1/2021.acl-long.413

[38] Yukun Ma, Erik Cambria, and Sa Gao. 2016. Label Embedding for Zero-shot Fine-grained Named Entity Typing. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. The COLING 2016 Organizing Committee, Osaka, Japan, 171–180. https://aclanthology.org/C16-1017

[39] Lorenzo Malandri, Fabio Mercorio, Mario Mezzanzanica, and Navid Nobani. 2021. TaxoRef: Embeddings Evaluation for AI-driven Taxonomy Refinement. In *Machine Learning and Knowledge Discovery in Databases. Research Track - European Conference, ECML PKDD 2021, Bilbao, Spain, September 13-17, 2021, Proceedings, Part III (Lecture Notes in Computer Science, Vol. 12977)*, Nuria Oliver, Fernando Pérez-Cruz, Stefan Kramer, Jesse Read, and José Antonio Lozano (Eds.). Springer, 612–627. https://doi.org/10.1007/978-3-030-86523-8_37

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

[41] Timothee Mickus, Denis Paperno, Mathieu Constant, and Kees van Deemter. 2020. What do you mean, BERT?. In *Proceedings of the Society for Computation in Linguistics 2020*. Association for Computational Linguistics, New York, New York, 279–290. https://aclanthology.org/2020.scil-1.35

[42] Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013. Linguistic Regularities in Continuous Space Word Representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Atlanta, Georgia, 746–751. https://aclanthology.org/N13-1090

[43] Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*. Association for Computational Linguistics, Suntec, Singapore, 1003–1011. https://aclanthology.org/P09-1113

[44] Kanishka Misra, Julia Rayz, and Allyson Ettinger. 2022. A Property Induction Framework for Neural Language Models. In *Proceedings of the 44th Annual Conference of the Cognitive Science Society*.

[45] Lorna I Morrow and M Frances Duffy. 2005. The representation of ontological category concepts as affected by healthy aging: Normative data and theoretical implications. *Behavior research methods* 37, 4 (2005), 608–625.

[46] Abhishek Panigrahi, Harsha Vardhan Simhadri, and Chiranjib Bhattacharyya. 2019. Word2Sense: Sparse Interpretable Word Embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 5692–5705. https://doi.org/10.18653/v1/P19-1570

[47] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Doha, Qatar, 1532–1543. https://doi.org/10.3115/v1/D14-1162

[48] Richard Socher, Milind Ganjoo, Hamsa Sridhar, Osbert Bastani, Christopher D. Manning, and Andrew Y. Ng. 2013. Zero-Shot Learning Through Cross-Modal Transfer. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). http://arxiv.org/abs/1301.3666

[49] Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. ConceptNet 5.5: An Open Multilingual Graph of General Knowledge. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, Satinder Singh and Shaul Markovitch (Eds.). AAAI Press, 4444–4451. http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14972

[50] Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R. Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R. Bowman, Dipanjan Das, and Ellie Pavlick. 2019. What do you learn from context? Probing for sentence structure in contextualized word representations. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net. https://openreview.net/forum?id=SJzSgnRcKX

[51] William Timkey and Marten van Schijndel. 2021. 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*. Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 4527–4546. https://doi.org/10.18653/v1/2021.emnlp-main.372

[52] Shikhar Vashishth, Manik Bhandari, Prateek Yadav, Piyush Rai, Chiranjib Bhattacharyya, and Partha Talukdar. 2019. Incorporating Syntactic and Semantic Information in Word Embeddings using Graph Convolutional Networks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 3308–3318. https://doi.org/10.18653/v1/P19-1320

[53] Nikhita Vedula, Patrick K. Nicholson, Deepak Ajwani, Sourav Dutta, Alessandra Sala, and Srinivasan Parthasarathy. 2018. Enriching Taxonomies With Functional Domain Knowledge. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*, Kevyn Collins-Thompson, Qiaozhu Mei, Brian D. Davison, Yiqun Liu, and Emine Yilmaz (Eds.). ACM, 745–754. https://doi.org/10.1145/3209978.3210000

[54] Ivan Vulić, Edoardo Maria Ponti, Anna Korhonen, and Goran Glavaš. 2021. LexFit: Lexical Fine-Tuning of Pretrained Language Models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, Online, 5269–5283. https://doi.org/10.18653/v1/2021.acl-long.410

[55] Ivan Vulić, Edoardo Maria Ponti, Robert Litschko, Goran Glavaš, and Anna Korhonen. 2020. Probing Pretrained Language Models for Lexical Semantics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Online, 7222–7240. https://doi.org/10.18653/v1/2020.emnlp-main.586

[56] Takashi Wada, Timothy Baldwin, Yuji Matsumoto, and Jey Han Lau. 2022. Unsupervised Lexical Substitution with Decontextualised Embeddings. In *Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022*, Nicoletta Calzolari, Chu-Ren Huang, Hansaem Kim, James Pustejovsky, Leo Wanner, Key-Sun Choi, Pum-Mo Ryu, Hsin-Hsi Chen, Lucia Donatelli, Heng Ji, Sadao Kurohashi, Patrizia Paggio, Nianwen Xue, Seokhwan Kim, Younggyun Hahm, Zhong He, Tony Kyungil Lee, Enrico Santus, Francis Bond, and Seung-Hoon Na (Eds.). International Committee on Computational Linguistics, 4172–4185. https://aclanthology.org/2022.coling-1.366

[57] Yile Wang, Leyang Cui, and Yue Zhang. 2021. Improving Skip-Gram Embeddings Using BERT. *IEEE ACM Trans. Audio Speech Lang. Process.* 29 (2021), 1318–1328.

[58] Chen Xing, Negar Rostamzadeh, Boris N. Oreshkin, and Pedro O. Pinheiro. 2019. Adaptive Cross-Modal Few-shot Learning. In *Proceedings of the Annual Conference on Neural Information Processing Systems*. 4848–4858.

[59] Wenhan Xiong, Jiawei Wu, Deren Lei, Mo Yu, Shiyu Chang, Xiaoxiao Guo, and William Yang Wang. 2019. Imposing Label-Relational Inductive Bias for Extremely Fine-Grained Entity Typing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, 773–784. https://doi.org/10.18653/v1/N19-1084

[60] Kun Yan, Zied Bouraoui, Ping Wang, Shoaib Jameel, and Steven Schockaert. 2021. Aligning Visual Prototypes with BERT Embeddings for Few-Shot Learning. In *Proceedings of the International Conference on Multimedia Retrieval*. 367–375.

[61] Kun Yan, Chenbin Zhang, Jun Hou, Ping Wang, Zied Bouraoui, Shoaib Jameel, and Steven Schockaert. 2022. 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*. AAAI Press, 2991–2999. https://ojs.aaai.org/index.php/AAAI/article/view/20205

[62] Jing Yao, Zhicheng Dou, and Ji-Rong Wen. 2022. Clarifying Ambiguous Keywords with Personal Word Embeddings for Personalized Search. *ACM Trans. Inf. Syst.* 40, 3 (2022), 43:1–43:29. https://doi.org/10.1145/3470564

[63] Xiaowei Zhao, Deqing Wang, Zhengyang Zhao, Wei Liu, Chenwei Lu, and Fuzhen Zhuang. 2021. A neural topic model with word vectors and entity vectors for short texts. *Inf. Process. Manag.* 58, 2 (2021), 102455. https://doi.org/10.1016/j.ipm.2020.102455