AMenDeD: Modelling Concepts by Aligning Mentions, Definitions and Decontextualised Embeddings

Amit Gajbhiye[1], Zied Bouraoui[2], Luis Espinosa Anke[1,3], Steven Schockaert[1]
[1]CardiffNLP, Cardiff University, UK, [2]CRIL CNRS & University of Artois, France, [3]AMPLYFI, UK
{gajbhiyea,espinosa-ankel,schockaerts1}@cardiff.ac.uk
zied.bouraoui@cril.fr

Abstract

Contextualised Language Models (LM) improve on traditional word embeddings by encoding the meaning of words in context. However, such models have also made it possible to learn high-quality decontextualised concept embeddings. Three main strategies for learning such embeddings have thus far been considered: (i) fine-tuning the LM to directly predict concept embeddings from the name of the concept itself, (ii) averaging contextualised representations of mentions of the concept in a corpus, and (iii) encoding definitions of the concept. As these strategies have complementary strengths and weaknesses, we propose to learn a unified embedding space in which all three types of representations can be integrated. We show that this allows us to outperform existing approaches in tasks such as ontology completion, which heavily depends on access to high-quality concept embeddings. We furthermore find that mentions and definitions are well-aligned in the resulting space, enabling tasks such as target sense verification, even without the need for any fine-tuning.

Keywords: lexical semantics, language models, concept embeddings, definitions

1. Introduction

The field of Natural Language Processing (NLP) has largely moved from static word embeddings to contextualised Language Models (LMs). However, static embeddings continue to play an important role in many applications. For instance, in few-shot and zero-shot learning, concept embeddings can provide prior knowledge about the considered labels (Xing et al., 2019; Yan et al., 2021; Luo et al., 2021; Huang et al., 2022; Ma et al., 2022; Li et al., 2023a). Concepts embeddings are similarly used in knowledge engineering, e.g. to help inform strategies for knowledge base completion (Vedula et al., 2018; Li et al., 2019; Malandri et al., 2021; Shi et al., 2023) or for aligning different resources (Trisedya et al., 2019; Kolyvakis et al., 2018). This has inspired a line of research dedicated to distilling high-quality concept embeddings from language models. We can distinguish three main strategies, which differ based on what is used as input: (i) only the name of the concept (Vulić et al., 2021; Liu et al., 2021a; Gajbhiye et al., 2022), (ii) mentions of the concept in a corpus (Liu et al., 2021b; Li et al., 2021, 2023b), or (iii) a definition (Mickus et al., 2022a; Ruzzetti et al., 2022). We refer to these strategies as concept name embeddings, concept mention embeddings and definition embeddings, respectively[1].

The three aforementioned strategies have complementary strengths and weaknesses. Concept name embeddings are efficient and relatively straightforward to train. However, since they rely on the knowledge that is captured by the LM itself, they are not suitable for modelling rare concepts. Moreover, they cannot distinguish between different senses of the same word. Concept mention embeddings are well-suited for modelling rare concepts but are harder to train. Moreover, not all mentions are equally informative. Accordingly, the performance of such concept mention embeddings depends on how the mentions are selected (Li et al., 2021; Wang et al., 2022). Finally, definition embedding models can distinguish different word senses and they are often available even for rare concepts (Ruzzetti et al., 2022). However, definitions typically only capture some of the knowledge that we may need, which means that the best results are often obtained by combining definition embeddings with other representations (Wang et al., 2022).

In this paper, we explore a simple technique for combining the three strategies. We start by training a standard concept name embedding model. We then train a concept mention embedding model based on the idea that a mention embedding should be similar to the embedding of the corresponding concept name. To prevent the model from simply re-learning the concept name embeddings, we mask the concept name when learning these mention embeddings. We train a definition embedding model in the same way. The concept name embedding model thus acts as an anchor which is used to align the three different representations.

This strategy has several important advantages. First, by using the concept name embeddings, we

[1] We publicly release the source code and data at https://github.com/amitgajbhiye/encoder_mentions.git

avoid the need for external resources when training the mention embedding model. Similarly, the concept name embedding model provides a natural supervision signal for training the definition embedding model. Second, by mapping mentions and concept names onto the same embedding space, we can naturally select the most informative mentions, by choosing those mentions whose embedding is most similar to that of the concept name. For ambiguous concepts, we can similarly select the mentions whose embedding is most similar to the embedding of the definition of the intended sense.

The paper is structured as follows. We first discuss related work in the next section. In Section 3 we then describe our strategy for learning concept representations. Finally, we present our experimental analysis in Section 4. The aims of this analysis are two-fold. First, we investigate to what extent mention and definition embeddings are aligned. In particular, we explore whether our approach is sufficient for learning mention embeddings that capture specific word senses, despite the fact that we have no sense-level supervision. Second, we explore the quality of the concept embeddings that can be obtained using our strategy. We use the task of ontology completion for this purpose. This is an important downstream application, which is highly reliant on having access to high-quality concept embeddings, and is thus a natural evaluation task.

2. Related Work

Learning Decontextualised Embeddings In recent years, two main strategies have emerged for learning decontextualised embeddings (i.e. static word vectors) using language models. First, and most straightforwardly, we can feed the word itself directly as the input to the language model. A word embedding can then be obtained, for instance, by averaging the embeddings of all tokens in the final layer of the pre-trained language model (Bommasani et al., 2020; Vulić et al., 2021; Gajbhiye et al., 2022). We will refer to such models as concept name encoders. Second, we can compute the contextualised representation of mentions of the word in some corpus, and aggregate the resulting mention embeddings, for instance by averaging them (Ethayarajh, 2019; Bommasani et al., 2020; Vulić et al., 2020). When relying on pre-trained language models, this latter strategy was found to clearly outperform the former (Bommasani et al., 2020). However, for both strategies, substantially better results can be obtained by fine-tuning the language model. For instance, Vulić et al. (2021) obtained significantly improved concept name embeddings after fine-tuning the LM on synonym and antonym pairs. Gajbhiye et al. (2022) instead considered the task of predicting commonsense properties, which involved jointly training the concept name encoder with a property encoder. MirrorBERT (Liu et al., 2021a) fine-tunes the LM using a self-supervision strategy, which also leads to improved concept name embeddings. Approaches based on aggregating mention embeddings similarly benefit from using fine-tuned LMs. MirrorWiC (Liu et al., 2021b) follows a self-supervision strategy for learning better contextualised representations, inspired by the MirrorBERT model. As another example, Li et al. (2023b) fine-tuned a mention encoder using distant supervision from ConceptNet[2].

Selecting Mentions One important question for methods that aggregate mention embeddings is how these mentions are selected. While most approaches simply use a random selection of mentions, Li et al. (2021) found that significantly better results can be obtained by filtering out mentions that reflect idiosyncratic usages of the corresponding word. Wang et al. (2022) analysed a number of strategies to select mentions in a more systematic way, for instance based on the structure of Wikipedia. Among others, they found it beneficial to include the definition of the word as one of the considered mentions. For concepts with a Wikipedia page, they also found that selecting mentions from that page outperformed randomly selecting mentions from across Wikipedia.

Exploiting Definitions Definitions are a natural resource to exploit for modelling word meaning. Accordingly, there is a long tradition of using definitions for improving word vectors, e.g. to learn representations of out-of-vocabulary terms (Bahdanau et al., 2017). Most closely related to this paper, there is work that links definitions to representative examples of word usage. For instance, Bevilacqua et al. (2020) have proposed the problem of generating definitions, given a word in context, which can then be used for tasks such as word sense disambiguation. Looking at the reverse problem, Barba et al. (2021) have studied the problem of generating representative examples of how a word is used in context, given its definition. The aforementioned works rely on sequence-to-sequence models and are not concerned with learning vector representations. The problem of aligning definition embeddings with word embeddings has been studied in the context of the reverse dictionary task. For instance Hill et al. (2016) train a definition encoder such that the embedding of a definition is similar to the embedding of the word being defined. Another relevant task is definition modelling (Noraset et al., 2017), which is concerned with generating a definition based on a given word embedding. Finally, Jo

[2] https://conceptnet.io

(2023) investigated how well the pretrained contextualised BERT representation of a word maps to its human-written definition and proposed a method to integrate definitions with such pretrained representations.

3. Learning Concept Embeddings

Our overall aim is to learn concept embeddings, which we view as compact representations of knowledge. We thus intuitively see this task as a form of knowledge distillation from LMs. In particular, we are interested in strategies for fine-tuning pretrained LMs to extract informative concept representations. We consider a number of LM encoders, which differ in the kind of input they receive. We first describe these encoders in Section 3.1 and then discuss the considered training strategies in Section 3.2. Finally, in Section 3.3 we describe strategies for obtaining concept embeddings from the trained encoders.

3.1. Encoders

We now describe the different encoders that we will rely on: a concept name encoder, a mention encoder and a definition encoder. Following Gajbhiye et al. (2022), we will also use a property encoder, which can be used to predict the semantic properties that are satisfied by concepts and which plays a central role in how they fine-tune their concept name encoder. We will rely as much as possible on existing encoding strategies: the novelty of our approach lies in how these different encoders are aligned and jointly used.

Concept Name Encoder The most straightforward strategy is to use the name of the concept as the input of a pre-trained language model. Previous work, however, has found that better results can be obtained by adding a short prompt. We will in particular rely on the strategy that was proposed by Gajbhiye et al. (2022), which uses a prompt of the following form: ⟨cls⟩ [CONCEPT] means ⟨mask⟩⟨sep⟩. As the embedding of the concept, we can then use the final-layer representation of the ⟨mask⟩ token. Let us write $\phi_{\text{con}}(c)$ for the resulting embedding of concept c. They also train a property encoder, which uses the same prompt, but is fine-tuned to represent properties (e.g. *red*) rather than concepts (e.g. *tomato*). We write $\phi_{\text{prop}}(p)$ for the embedding of property p.

Mention Encoder The aim of the mention encoder is to learn concept embeddings from sentences that mention these concepts. Following Li et al. (2021), to encode a given mention, we mask the occurrence of the concept, feed the resulting sentence to an LM and use the representation of the <mask> token as the corresponding embedding. Li et al. (2021) found that masking the concept leads to embeddings that are more predictive of the properties that are satisfied by concepts (although the resulting embeddings were also found to be less suitable for modelling word similarity). In our setting, masking also has the benefit that the resulting embeddings should be more complementary to those that are obtained from the concept name encoder: by masking the concept name, the encoder is forced to focus on modelling the context in which the concept occurs. We write $\phi_{\text{men}}(m)$ for the embedding of a mention m, where a *mention* corresponds to a sentence in which some target concept is highlighted (i.e. the concept to be masked).

Definition Encoder We also consider an encoder which learns concept embeddings from definitions. In this case, we use a prompt of the form <mask>: [DEFINITION] and use the contextualised (i.e. final-layer) representation of the <mask> token. Note that this encoder has the same form as the mention encoder. For this encoder, we assume that the definitions themselves do not mention the name of the concept. This ensures that the encoder captures the actual definition, and is thus complementary to the concept name encoder. We write $\phi_{\text{def}}(d)$ for the embedding of a definition d.

3.2. Training Strategies

Our starting point is the model from Gajbhiye et al. (2022), where a concept name encoder and property encoder are jointly trained as a bi-encoder, using a large set of (concept, property) examples. Specifically, they jointly train both encoders using binary cross-entropy (BCE):

$$\begin{aligned}\mathcal{L}^{\text{bce}}_{\text{name}} = &-\sum_{(c,p)\in X^+} \log \sigma(\phi_{\text{con}}(c)\cdot\phi_{\text{prop}}(p)) \\ &-\sum_{(c,p)\in X^-} \log(1-\sigma(\phi_{\text{con}}(c)\cdot\phi_{\text{prop}}(p)))\end{aligned}$$

where X^+ is a set of positive examples, i.e. $(c,p)\in X^+$ means that c is assumed to have property p, and X^- is a set of in-batch negative examples.

Concept Name Embeddings as Anchors We start by fine-tuning the concept name and property encoders. The resulting concept name encoder can only encode knowledge that is captured by the LM itself, hence it may not be suitable for rare concepts. However, it can provide us with high-quality representations for well-known concepts. Based on this view, we propose to use the representations of the concept name encoder as anchors for training the mention and definition encoders. For instance, we can train the mention encoder by requiring that mention embeddings are similar to the corresponding concept name embeddings:

$$\mathcal{L}^{\text{bce}}_{\text{men}} = -\sum_{(c,m)\in M^+} \log \sigma(\phi_{\text{con}}(c)\cdot\phi_{\text{men}}(m)) \\ - \sum_{(c,m)\in M^-} \log(1-\sigma(\phi_{\text{con}}(c)\cdot\phi_{\text{men}}(m)))$$

where the concept name encoder ϕ_{con} is frozen. Here $(c,m)\in M^+$ means that m is a mention of the concept c, whereas the set of negative examples M^- consists of pairs (c,m) where m is a mention of a different concept. We will also experiment with a variant that instead relies on the InfoNCE contrastive loss (van den Oord et al., 2018), given its strong performance across a wide range of representation learning tasks. We will refer to this variant as $\mathcal{L}^{\text{info}}_{\text{men}}$:

$$-\sum_{(c,m)\in M^+} \log \frac{\exp(\cos(\phi_{\text{con}}(c),\phi_{\text{men}}(m))/\tau)}{\sum_{m'}\exp(\cos(\phi_{\text{con}}(c),\phi_{\text{men}}(m'))/\tau)}$$

where $\tau > 0$ is the temperature hyperparameter, the summation in the denominator ranges over all mentions in the given mini-batch (across all concepts), and we again assume that the concept name encoder ϕ_{con} is frozen. Finally, the definition encoder is trained in the same way. Let us refer to the corresponding loss as $\mathcal{L}^{\text{bce}}_{\text{def}}$ or $\mathcal{L}^{\text{info}}_{\text{def}}$, depending on whether BCE or InfoNCE is used.

Joint Learning The aforementioned training strategy is based on the assumption that the available (concept, property) training examples are sufficient for training a high-quality concept name encoder. As an alternative, we also experiment with a strategy in which all encoders are jointly trained. For instance, when using BCE, we can use the following overall loss function:

$$\mathcal{L} = \mathcal{L}^{\text{bce}}_{\text{name}} + \mathcal{L}^{\text{bce}}_{\text{men}} + \mathcal{L}^{\text{bce}}_{\text{def}}$$

In this case, the concept name encoder is still used for training the mention and definition encoder, but it is no longer frozen during this process.

3.3. Concept Embedding Strategies

After the encoders from Section 3.2 have been trained, we can consider several strategies for learning a final concept vector. Let c be the name of the concept that we want to represent. Let us furthermore assume that we have access to a set of mentions M_c of concept c. Let us write $\mathbf{c}$ for the concept embedding that we want to learn. We will experiment with the following variants:

Name We use the concept name embedding, i.e. $\mathbf{c} = \phi_{\text{con}}(c)$.

Men We use the average of the mention embeddings, i.e. $\mathbf{c} = \frac{1}{|M_c|}\sum_{m\in M_c}\phi_{\text{men}}(m)$.

Men$_k$ We first determine the k mention vectors m whose embedding $\phi_{\text{men}}(m)$ is most similar to the concept name embedding $\phi_{\text{con}}(c)$, in terms of cosine similarity. Let us write these mentions as $m_1, ..., m_k$. We represent the concept as $\mathbf{c} = \frac{1}{k}\sum_{i=1}^{k}\phi_{\text{men}}(m_i)$.

In applications where we have access to definitions, we can use definition embeddings as an alternative to the concept name embeddings, which would help to select mentions where the target word is used with its intended sense.

4. Experiments

Our analysis focuses on the following research questions:

- What is the best strategy for training the different encoders? Is it beneficial to use the concept name encoder as an anchor, or is it better to jointly train the different encoders?
- How well are the mention encoder and concept name encoder aligned? Can we select sense-specific mentions of a word by comparing mention embeddings with definition embeddings?
- How successful is the overall approach in learning informative concept embeddings.

We refer to our method as AMenDeD (Aligning Mentions, Definitions and Decontextualised embeddings). In Section 4.1, we first focus on tasks which involve aligning mentions and definitions. Section 4.2 then focuses on the quality of different strategies for learning concept embeddings, which we evaluate on the downstream task of ontology completion. Finally, Section 4.3 presents a qualitative analysis. We first provide some details about our experimental set-up.

Datasets To train the bi-encoder, i.e. the concept name and property encoders, we need sets of (concept, property) pairs. We consider three such sets:

- We consider the 100K (concept, property) pairs from Microsoft Concept Graph (Ji et al., 2019) that were used by Gajbhiye et al. (2022). Note that in this case, the properties are in fact hypernyms. However, Gajbhiye et al. (2022) found training on this dataset to be useful because many of the hypernyms refer to semantic properties (e.g. *low-sugar berry*).
- We consider a dataset of 109K (concept, property) pairs that was collected by Chatterjee et al. (2023) using ChatGPT.

- We consider (concept, property) pairs from ConceptNet. Specifically, we converted instances of the relations IsA, PartOf, LocatedAt, UsedFor and HasProperty, leading to an additional set of 63K (concept, property) pairs.

As our default choice, we rely on the combination of the ConceptNet and ChatGPT datasets (CN+Chat). We found that combining these two datasets consistently outperformed using either dataset alone. As an alternative, we also experiment with training the encoders on the combination of all three datasets (MS+CN+Chat). To train the mention encoder, we need to decide on a vocabulary of concepts, and we need a strategy for collecting mentions of these concepts. We consider two possibilities:

- We use the 5098 concepts that appear in the CN+Chat training set. For each of these concepts, we randomly selected 100 mentions from Wikipedia. Sentences with a length greater than 32 were filtered out. On average, we ended up with 65 mentions per concept.

- We train the model using dictionary examples. This is motivated by the fact that such examples are often carefully chosen to be informative and representative of how the target word is typically used. We rely in particular on the 3D-EX resource from Almeman et al. (2023), which aggregates a number of lexical resources. However, we omit examples from the Urban Dictionary and Sci-Definition, as they are less representative. The resulting resource covers 224K unique concepts, with an average of 1.3 mentions per concept.

We refer to these training sets as *Wiki* and *3D*. We will also consider a model that is trained on the combination of both. Finally, to train the definition encoder, we have relied on WordNet definitions[3] and the CODWOE dataset from Semeval-2022 Task 1 (Mickus et al., 2022b).

Training Details We train the mention and definition encoders with a batch size of 32. We use the Adam optimiser with an initial learning rate of $2e-6$ and a cosine learning rate warm-up over 20$\%$ of the training data. The encoders are trained with a maximum of 100 epochs with an early stopping patience of 3. For encoders trained with InfoNCE we set the temperature τ to 0.05.

4.1. Aligning Mentions and Definitions

One advantage of our strategy is that mention and definition embeddings are aligned. In this section, we consider two intrinsic benchmarks which directly evaluate this aspect: CoDA21 (Senel et al., 2022) and WiC-TSV (Breit et al., 2021).

CoDA21 Each CoDA21 problem instance focuses on k words. For each of these words, there is one masked mention (i.e. a sentence from a corpus that mentioned the word, but where the word itself was masked) and one definition (which does not mention the word itself either). The task is to align the k mentions with the k definitions, i.e. to predict for each of the mentions which is the corresponding definition. Senel et al. (2022) solve this task by assigning a score to each of the $k!$ possible alignments, and then simply selecting the alignment with the highest score. This score is defined as the sum of the scores of the k corresponding mention-definition pairings. To score a given mention-definition pairing, their main baseline relies on pre-trained language models. Specifically, given the mention m, they replace the mask by a made-up word x. Writing m_x for the resulting sentence, they use the following prompt "m_x `Definition of` x `is`" and then they evaluate the log of the probability that the definition follows this prefix. To use our models, we simply score a given mention-definition pairing as the cosine similarity between the corresponding mention and definition embeddings. Note that this is an unsupervised task, i.e. there is no training set associated with CoDA21. The test set is organised in different partitions: clean-hard, clean-easy, noisy-hard, noisy-easy. Here, clean refers to the fact that the mentions are selected to be informative about the target word, while hard refers to the fact that the k words are chosen to be closely related co-hyponyms. Following Senel et al. (2022) we separately report results for nouns (N) and verbs (V) and use accuracy as the evaluation metric.

The results are reported in Table 1. We report variants of our method which differ in which loss function was used for training the mention and definition encoders (InfoNCE or BCE). In most configurations, we use the bi-encoder as a static anchor, but we also experiment with the joint learning strategy, where the bi-encoder is trained jointly with the mention and definition encoders. These configurations are referred to as InfoNCE$_{\text{joint}}$ and BCE$_{\text{joint}}$, depending on whether InfoNCE or BCE was used for training the mention and definition encoders[4]. The configurations in Table 1 also differ in the training data that was used for the bi-encoder and the training data that was used for the mention encoder.

A number of clear observations can be made. First, all of the considered variants of our approach substantially outperform the baselines. While these

[3] `https://huggingface.co/datasets/marksverdhei/wordnet-definitions-en-2021`

[4] Note that we always use BCE for training the bi-encoder. We experimented with using InfoNCE for the bi-encoder as well, but initial results were not promising.

Model	**Loss**	**Bi-encoder**	**Mentions**	**clean-hard**		**clean-easy**		**noisy-hard**		**noisy-easy**	
				N	**V**	**N**	**V**	**N**	**V**	**N**	**V**
$\text{BERT}^*_{\text{large}}$				22	22	19	21	19	20	20	20
$\text{RoBERTa}^*_{\text{large}}$				26	30	30	30	27	29	30	33
$\text{GPT2}^*_{\text{large}}$				38	34	47	42	39	37	46	41
$\text{GPT2}^*_{\text{XL}}$				42	36	49	42	40	36	46	43
AMenDeD	InfoNCE	CN+Chat	Wiki	63	62	**77**	72	54	55	64	62
AMenDeD	InfoNCE	MS+CN+Chat	Wiki	63	65	75	71	54	55	64	61
AMenDeD	BCE	CN+ChatGPT	Wiki	**68**	65	76	**73**	**56**	57	**65**	62
AMenDeD	$\text{InfoNCE}_{\text{joint}}$	CN+Chat	Wiki	65	**66**	74	71	53	**58**	63	**63**
AMenDeD	$\text{BCE}_{\text{joint}}$	CN+Chat	Wiki	59	65	74	69	55	56	63	60
AMenDeD	InfoNCE	CN+Chat	3D	62	60	74	68	54	54	62	60
AMenDeD	InfoNCE	CN+Chat	3D+Wiki	62	62	73	71	53	55	62	61

Table 1: Results on CoDA21 in terms of accuracy (%). *Baseline results are taken from Senel et al. (2022).

Model	**FT**	**Loss**	**Bi-encoder**	**Mentions**	**WNT/WKT**	**CTL**	**MESH**	**CS**	**All**
U-dBERT*	no				49.2	57.4	62.0	63.1	56.9
U-BERT*	no				51.5	56.9	65.9	69.0	54.4
$\text{BERT}^*_{\text{large}}$	yes				**77.1**	73.1	75.3	70.2	75.3
GlossBERT*	yes				75.7	**75.5**	74.1	79.8	**76.0**
$\text{GlossBERT}^*_{\text{ws}}$	yes				75.2	70.4	**78.5**	**82.7**	75.9
AMenDeD	no	InfoNCE	CN+Chat	Wiki	72.4	64.4	64.8	76.2	71.7
AMenDeD	no	InfoNCE	MS+CN+Chat	Wiki	67.6	62.4	52.8	57.7	64.6
AMenDeD	no	BCE	CN+ChatGPT	Wiki	67.8	63.9	59.7	73.8	65.5
AMenDeD	no	$\text{InfoNCE}_{\text{joint}}$	CN+Chat	Wiki	68.5	61.5	51.4	72.0	66.2
AMenDeD	no	$\text{BCE}_{\text{joint}}$	CN+Chat	Wiki	59.4	55.6	50.5	61.3	58.6
AMenDeD	no	InfoNCE	CN+Chat	3D	$74.2^\dagger$	70.2	57.4	69.6	$71.5^\dagger$
AMenDeD	no	InfoNCE	CN+Chat	3D+Wiki	$73.5^\dagger$	64.9	73.2	77.4	$73.3^\dagger$

Table 2: Results on WiC-TSV in terms of accuracy (%). *Baseline results are taken from Breit et al. (2021). †These results likely overestimate the performance of the corresponding models, since the 3D-EX pretraining-data includes examples from Wordnet and Wiktionary.

baselines are pre-trained LMs and our models were specifically trained on definitions and mentions, it is nonetheless interesting to see that our fine-tuned BERT-large model can significantly outperform even the 1.5B parameter GPT2_{XL} model. Another interesting observation is that our models achieve a similar performance for nouns and verbs. This is surprising, given that the training data almost exclusively consists of nouns. Finally, we can see that each of the considered variants performs rather similarly on this benchmark.

WiC-TSV In the case of WiC-TSV, the problem of mention-definition alignment is treated as a binary classification problem: given a mention of a word in context and a definition of that word, the task is to predict whether the definition captures the correct sense of the word in context. This task is called Target Sense Verification. The dataset includes test instances from four different domains: general terms from WordNet and Wiktionary (WNT/WKT), cocktails (CTL), medical terms obtained from MeSH, and computer science terms (CS). Following Breit et al. (2021), we also report the results across the entire dataset, covering all domains. Note that 3D-EX covers WordNet and Wiktionary, hence we cannot fairly evaluate the variants of our models that use 3D-EX on the WNT/WKT test instances. The baselines reported by Breit et al. (2021) feed the concatenation of the mention and the definition to an LM. Then they concatenate the contextualised embeddings of the [CLS] token and the word in context, as well as the average embedding of the tokens from the definition, and feed the resulting vector to a linear classifier. As language models, they considered the standard BERT models and GlossBERT (Huang et al., 2019), a specialised BERT model for word sense disambiguation. A variant of GlossBERT with a form of weak supervision was also considered ($\text{GlossBERT}_{\text{ws}}$). Beyond these supervised models, they also evaluated two unsupervised models. In this case, they obtained

		Wine	Econ	Olym	Tran	SUMO
	GloVe*	14.2	14.1	9.9	8.3	34.9
	Skipgram*	13.8	13.5	8.3	7.2	33.4
	Numberbatch*	25.6	26.2	26.8	16.0	47.3
	MirrorBERT*	22.5	23.8	20.9	12.7	40.1
	MirrorWiC*	24.7	24.9	22.1	13.9	46.9
	ConCN*	29.1	31.3	27.6	19.7	50.4
	ConCN + filt.*	31.3	32.4	29.7	20.9	52.6
	Name	30.8	30.5	28.6	19.8	51.3
Wiki	Men	29.2	28.3	27.1	18.5	45.3
	Men + filt.	30.9	29.1	27.5	19.8	50.7
	Men$_k$	31.5	29.5	29.4	20.3	51.7
	Men$_k$ + filt.	31.9	33.2	30.4	21.7	52.9
3D	Men	31.9	32.5	29.5	20.8	51.8
	Men + filt.	33.7	33.2	30.3	21.3	52.7
	Men$_k$	32.1	32.9	29.7	20.9	52.3
	Men$_k$ + filt.	36.8	34.2	31.9	22.1	53.1
3D+Wiki	Men	32.1	32.7	29.9	21.1	52.4
	Men + filt.	34.6	33.3	30.5	21.9	53.2
	Men$_k$	33.8	33.1	30.3	21.3	52.8
	Men$_k$ + filt.	**39.5**	**35.8**	**32.2**	**22.5**	**53.5**

Table 3: Results for ontology completion in terms of F1 (%). *Baseline results were taken from Li et al. (2023b).

embeddings for the word in context and the definition using a pre-trained BERT model and then classify a problem instance as positive if their cosine similarity is above some threshold. Two unsupervised models were evaluated, which use BERT and DistilBERT respectively. Note that this threshold is tuned on the validation data, which means that the methods are, in fact, not fully unsupervised. We will therefore refer to them as variants without fine-tuning rather than unsupervised variants. We evaluate our variants without fine-tuning, as our focus is on assessing to what extent our mention and definition encoders are aligned. In particular, we simply obtain the mention and definition embeddings using our pre-trained encoders and classify an example as positive if the cosine similarity is sufficiently high, by again tuning the threshold on the validation data.

The results are summarised in Table 2. They show that our default configuration (InfoNCE loss, ConceptNet + ChatGPT training, Wikipedia mentions) performs well overall, outperforming all variants which do not rely on 3D-EX. Variants that were pre-trained on 3D-EX can only be fairly evaluated on CTL, MESH and CS. The results on these domains suggest that including mentions from 3D-EX is indeed useful. Overall, our models perform quite well, substantially outperforming the baselines without fine-tuning. While our models have been trained on definitions and mentions, it should be noted that they have not been trained on definition-mention pairings, and in particular that they have not been trained on any sense-level supervision. The results in Table 2 show that our mention encoder has nonetheless learned to encode word senses.

4.2. Learning Concept Embeddings

As an example of a downstream task, we consider the problem of ontology completion, which has already been used for evaluating mention embeddings in previous work (Li et al., 2021, 2023b). This task consists in predicting missing concept inclusions in description logic ontologies. Such ontologies contain basic concept inclusions such as $\textit{Banana} \sqsubseteq \textit{Fruit}$, which essentially encode hypernyms, but also concept inclusions involving logical connectives and quantifiers. For instance, an inclusion such as $\textit{Author} \sqsubseteq \exists \textit{hasPublished}.\textit{Article}$ expresses that an author is someone who has published an article, while $\textit{Female} \sqcap (\exists \textit{hasChild}.\top) \sqsubseteq \textit{Mother}$ expresses that a female person who has a child is a mother. Li et al. (2019) proposed a framework for predicting plausible concept inclusions using a graph convolutional network. The nodes of this network correspond to the concepts from a given ontology, and the input representations correspond to pre-trained concept embeddings. The quality of the predicted concept inclusions strongly depends on the quality of these pre-trained embeddings, which makes this a suitable task for evaluating concept representations. The problem is formalised as a binary classification problem, with the results reported in terms of F1 score.

We consider the following baselines. First, *GloVe* (Pennington et al., 2014), *Skipgram* (Mikolov et al., 2013) and *Numberbatch* (Speer et al., 2017) are traditional static word embedding models. *MirrorBERT* (Liu et al., 2021a) is a contrastively fine-tuned BERT model, aimed at learning high-quality concept and sentence embeddings. Next, we also consider two approaches that are based on aggregating contextualised embeddings of concept mentions: *MirrorWiC* (Liu et al., 2021b), which is a variant of *MirrorBERT* aimed at modelling words in context, and *ConCN* (Li et al., 2023b), which uses a mention encoder that was trained on a distant supervision signal based on ConceptNet. We also report the results of a variant of *ConCN* which uses a strategy for filtering mention vectors proposed by Li et al. (2021) (*ConCN + filt.*). This latter variant is the current state-of-the-art. Their filtering strategy identifies mentions that are likely to capture idiosyncratic properties, by looking at the nearest neighbours of each mention embedding. Specifically, if all the nearest neighbours correspond to mentions of the same word, a mention is deemed to be idiosyncratic and omitted.

Concept	Nearest mentions
doorknob	Afterwards, the exterior double doors on that building were changed so they only had one doorknob, and this remains today.
airplane	He was the sixth person to make a successful flight over the Atlantic Ocean with a single engine, single seat airplane.
axe	One Christmas, the town gives Paul a double-bladed axe to help chop down timber.
colander	A colander is a kitchen utensil for draining food.
ashtray	An ashtray is a receptacle for ash from cigarettes and cigars
crayon	Common tools are graphite pencils, pen and ink, inked brushes, wax color pencils, crayons, charcoals, pastels, and markers.
donkey	Traditional pack animals are diverse including camels, goats, yaks, reindeer, water buffaloes, and llamas as well as the more familiar pack animals like dogs, horses, donkeys, and mules.
dagger	The acinaces, also transliterated as akinakes or akinaka is a type of dagger or xiphos (short sword) used mainly in the first millennium BCE in the eastern Mediterranean Basin, especially by the Medes, Scythians, Persians and Caspians, then by the Greeks.

Table 4: Sentences corresponding to the most similar mention vector, for a number of common concepts.

We consider different variants of our approach. We experiment with mentions encoders that were trained on Wikipedia mentions, 3D-EX, and on the combination of both. For these experiments, we use the bi-encoder that was trained on ConceptNet + ChatGPT, and we use InfoNCE for training the mention encoder, as this configuration was most effective in the WiC-TSV experiment. To learn concept embeddings, we compare the Name, Men and Men$_k$ strategies. For the Men$_k$ representations, we set $k = 5$. We also experiment with the filtering strategy from Li et al. (2021), either as an alternative to the Men$_k$ filtering strategy, or in addition to it. In the latter case, we first apply the filtering strategy from Li et al. (2021) and then select the 5 remaining mentions whose embedding is closest to the concept name embedding. We refer to these variants as *Men + filt.* and *Men$_k$ + filt.*

The results are summarised in Table 3, where we use the same five ontologies as Li et al. (2019). We also rely on their framework for solving the task, only changing the concept embeddings. First, we can see that *Name* outperforms all methods that are not based on aggregating mention embeddings. In particular, *Name* clearly outperforms *MirrorBERT*, which shows the limitations of self-supervised training for learning concept embeddings. The performance of *Men* is highly dependent on the training set. For *Wiki*, we find that *Men* underperforms both *Name* and the ConCN baseline (except for *Wine*). However, when the mention encoder is pre-trained on 3D-EX, *Men* performs considerably better, while the best results are obtained for the variant that was trained on both 3D-EX and Wikipedia. We can furthermore see that both the filtering strategy from Li et al. (2021) and the *Men$_k$* filtering strategy are highly effective, consistently improving the results for all ontologies and all configurations. Moreover, these two filtering strategies are complementary: the combined strategy *Men$_k$ + filt.* substantially outperforms all other variants.

4.3. Qualitative Analysis

As we saw in Section 4.2, the ability to select informative mentions is an important advantage of our approach. Table 4 shows, for a number of concepts, the sentence whose corresponding mention vector is most similar to the concept name embedding. As can be seen, the selected sentences are informative in different ways. First, some sentences reveal specific properties of the concepts. For instance, the sentence for *doorknob* reveals that it is part of a door, while the sentence for *airplane* reveals that this is something which is used for flight and the sentence for *axe* reveals that it is used for chopping. Second, as the examples for *colander* and *ashtray* show, sometimes a definition of the concept is selected. Next, some sentences mention the target concept among a list of co-hyponyms, which is illustrated for the concepts *crayon* and *donkey*. Finally, the sentence that was identified for *dagger* suggests that this concept is similar to a short sword.

5. Conclusions

In this paper, we have proposed a strategy for training mention and definition encoders, by using a concept name embedding model as an anchor. We found that the resulting mention encoder allows us to learn concept representations that substantially outperform the state-of-the-art in ontology completion. Our framework addresses a key problem when distilling concept embeddings from language models: approaches based on mention encoders tend to give the best results, but it is harder to fine-tune mention encoders in a meaningful way. Our proposed solution is conceptually straightforward and easy to use. In our analysis, we furthermore found that the mention embeddings are aligned with definitions of the corresponding word senses, despite the fact that no sense-level supervision was provided to these models. This makes it possible to learn sense-specific concept embeddings, something which is not possible with most existing strategies for learning concept embeddings.

Acknowledgments This work was supported by EPSRC grant EP/V025961/1, ANR-22-CE23-0002 ERIANA and HPC resources from GENCI-IDRIS (Grant 2023-[AD011013338R1]).

6. Bibliographical References

Fatemah Almeman, Hadi Sheikhi, and Luis Espinosa Anke. 2023. 3d-ex : A unified dataset of definitions and dictionary examples. *CoRR*, abs/2308.03043.

Dzmitry Bahdanau, Tom Bosc, Stanislaw Jastrzebski, Edward Grefenstette, Pascal Vincent, and Yoshua Bengio. 2017. Learning to compute word embeddings on the fly. *CoRR*, abs/1706.00286.

Edoardo Barba, Luigi Procopio, Caterina Lacerra, Tommaso Pasini, and Roberto Navigli. 2021. Exemplification modeling: Can you give me an example, please? In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*, pages 3779–3785. ijcai.org.

Michele Bevilacqua, Marco Maru, and Roberto Navigli. 2020. Generationary or "how we went beyond word sense inventories and learned to gloss". In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7207–7221, Online. Association for Computational Linguistics.

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*, pages 4758–4781, Online. Association for Computational Linguistics.

Anna Breit, Artem Revenko, Kiamehr Rezaee, Mohammad Taher Pilehvar, and Jose Camacho-Collados. 2021. WiC-TSV: An evaluation benchmark for target sense verification of words in context. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1635–1645, Online. Association for Computational Linguistics.

Usashi Chatterjee, Amit Gajbhiye, and Steven Schockaert. 2023. Cabbage sweeter than cake? analysing the potential of large language models for learning conceptual spaces. *arXiv preprint arXiv:2310.05481*.

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)*, pages 55–65, Hong Kong, China. Association for Computational Linguistics.

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*, pages 3971–3983, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Felix Hill, Kyunghyun Cho, Anna Korhonen, and Yoshua Bengio. 2016. Learning to understand phrases by embedding the dictionary. *Transactions of the Association for Computational Linguistics*, 4:17–30.

James Y. Huang, Bangzheng Li, Jiashu Xu, and Muhao Chen. 2022. Unified semantic typing with meaningful label inference. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2642–2654, Seattle, United States. Association for Computational Linguistics.

Luyao Huang, Chi Sun, Xipeng Qiu, and Xuanjing Huang. 2019. GlossBERT: BERT for word sense disambiguation with gloss knowledge. 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)*, pages 3509–3514, Hong Kong, China. Association for Computational Linguistics.

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):238–270.

Hwiyeol Jo. 2023. A self-supervised integration method of pretrained language models and word definitions. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 14–26. Association for Computational Linguistics.

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)*, pages 787–798, New Orleans, Louisiana. Association for Computational Linguistics.

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*, pages 3850–3856. ijcai.org.

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*, volume 11778 of *Lecture Notes in Computer Science*, pages 435–452. Springer.

Na Li, Zied Bouraoui, and Steven Schockaert. 2023a. Ultra-fine entity typing with prior knowledge about labels: A simple clustering based strategy. *CoRR*, abs/2305.12802.

Na Li, Hanane Kteich, Zied Bouraoui, and Steven Schockaert. 2023b. 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 2023, Taipei, Taiwan, July 23-27, 2023*, pages 216–226. ACM.

Fangyu Liu, Ivan Vulić, Anna Korhonen, and Nigel Collier. 2021a. 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*, pages 1442–1459, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Qianchu Liu, Fangyu Liu, Nigel Collier, Anna Korhonen, and Ivan Vulić. 2021b. MirrorWiC: On eliciting word-in-context representations from pretrained language models. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 562–574, Online. Association for Computational Linguistics.

Qiaoyang Luo, Lingqiao Liu, Yuhao Lin, and Wei Zhang. 2021. Don't miss the labels: Label-semantic augmented meta-learner for few-shot text classification. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2773–2782, Online. Association for Computational Linguistics.

Jie Ma, Miguel Ballesteros, Srikanth Doss, Rishita Anubhai, Sunil Mallya, Yaser Al-Onaizan, and Dan Roth. 2022. Label semantics for few shot named entity recognition. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1956–1971, Dublin, Ireland. Association for Computational Linguistics.

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*, volume 12977 of *Lecture Notes in Computer Science*, pages 612–627. Springer.

Timothee Mickus, Kees Van Deemter, Mathieu Constant, and Denis Paperno. 2022a. Semeval-2022 task 1: CODWOE – comparing dictionaries and word embeddings. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 1–14, Seattle, United States. Association for Computational Linguistics.

Timothee Mickus, Kees van Deemter, Mathieu Constant, and Denis Paperno. 2022b. Semeval-2022 task 1: CODWOE - comparing dictionaries and word embeddings. In *Proceedings of the 16th International Workshop on Semantic Evaluation, SemEval@NAACL 2022, Seattle, Washington, United States, July 14-15, 2022*, pages 1–14. Association for Computational Linguistics.

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*, pages 746–751, Atlanta, Georgia. Association for Computational Linguistics.

Thanapon Noraset, Chen Liang, Larry Birnbaum, and Doug Downey. 2017. Definition modeling: Learning to define word embeddings in natural language. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 3259–3266. AAAI Press.

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)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Elena Sofia Ruzzetti, Leonardo Ranaldi, Michele Mastromattei, Francesca Fallucchi, Noemi Scarpato, and Fabio Massimo Zanzotto. 2022. Lacking the embedding of a word? look it up into a traditional dictionary. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2651–2662, Dublin, Ireland. Association for Computational Linguistics.

Lütfi Kerem Senel, Timo Schick, and Hinrich Schuetze. 2022. CoDA21: Evaluating language understanding capabilities of NLP models with context-definition alignment. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 815–824, Dublin, Ireland. Association for Computational Linguistics.

Jingchuan Shi, Jiaoyan Chen, Hang Dong, Ishita Khan, Lizzie Liang, Qunzhi Zhou, Zhe Wu, and Ian Horrocks. 2023. Subsumption prediction for e-commerce taxonomies. In *The Semantic Web - 20th International Conference, ESWC 2023, Hersonissos, Crete, Greece, May 28 - June 1, 2023, Proceedings*, volume 13870 of *Lecture Notes in Computer Science*, pages 244–261. Springer.

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*, pages 4444–4451. AAAI Press.

Bayu Distiawan Trisedya, Jianzhong Qi, and Rui Zhang. 2019. Entity alignment between knowledge graphs using attribute embeddings. 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 297–304. AAAI Press.

Aäron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748.

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*, pages 745–754. ACM.

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)*, pages 5269–5283, Online. Association for Computational Linguistics.

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)*, pages 7222–7240, Online. Association for Computational Linguistics.

Yixiao Wang, Zied Bouraoui, Luis Espinosa Anke, and Steven Schockaert. 2022. Sentence selection strategies for distilling word embeddings from BERT. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 2591–2600, Marseille, France. European Language Resources Association.

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*, pages 4848–4858.

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*, pages 367–375.