Learning Sense-Specific Static Embeddings using Contextualised Word Embeddings as a Proxy

Yi Zhou
Department of Computer Science
University of Liverpool, UK
Y.Zhou71@liverpool.ac.uk

Danushka Bollegala
Department of Computer Science
University of Liverpool, UK
danushka@liverpool.ac.uk

Abstract

Contextualised word embeddings generated from Neural Language Models (NLMs), such as BERT, represent a word with a vector that considers the semantics of the target word as well its context. On the other hand, static word embeddings such as GloVe represent words by relatively low-dimensional, memory- and compute-efficient vectors but are not sensitive to the different senses of the word. We propose Context Derived Embeddings of Senses (CDES), a method that extracts sense related information from contextualised embeddings and injects it into static embeddings to create sense-specific static embeddings. Experimental results on multiple benchmarks for word sense disambiguation and sense discrimination tasks show that CDES can accurately learn sense-specific static embeddings reporting comparable performance to the current state-of-the-art sense embeddings.
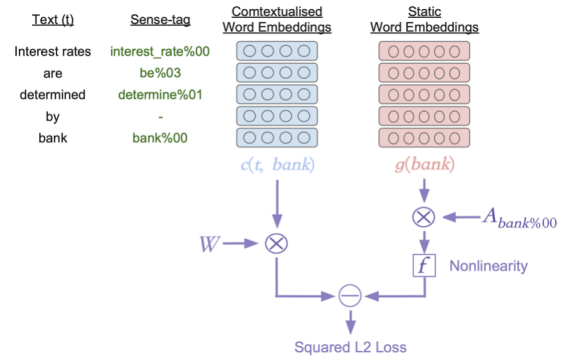
Figure 1: Outline of CDES. Given a sense-tagged sentence t, we compute a sense embedding for the ambiguous word *bank* by multiplying its static word embedding, $g(bank)$, by a sense-specific projection matrix, $\mathbf{A}_{bank\%00}$, corresponding to the correct sense of the word. Projection matrices are learnt by minimising the squared ℓ_2 loss between the linearly transformed (via a matrix $\mathbf{W}$) contextualised embedding, $c(t, bank)$, and of the (nonlinearly transformed via function f) sense embedding of *bank*.

1 Introduction

Representing the meanings of words using low-dimensional vector embeddings has become a standard technique in NLP. *Static word embeddings* (Mikolov et al., 2013; Pennington et al., 2014) represent words at the *form* level by assigning a single vector for all occurrences of a word irrespective of its *senses*. However, representing ambiguous words such as *bass*, which could mean either a *musical instrument* or a type of *fish*, using a single embedding is problematic.

To address this problem, *sense-specific static word embedding* methods (Reisinger and Mooney, 2010; Neelakantan et al., 2014; Huang et al., 2012) assign multiple embeddings to a single polysemous word corresponding to its multiple senses. However, these embeddings are context-insensitive and we must resort to different heuristics such as selecting the sense embedding of the ambiguous word that is most similar to the context, to determine which embedding should be selected to represent the word.

On the other hand, *contextualised word embeddings* generated from NLMs (Peters et al., 2018a; Devlin et al., 2019; Liu et al., 2019) represent a word in a given context by an embedding that considers both the meaning of the word itself as well as its

context. Different types of information such as word sense, dependency, and numeracy have shown to be encoded in contextualised word embeddings, providing rich, context-sensitive input representations for numerous downstream NLP applications. More recently, Loureiro and Jorge (2019a) and Scarlini et al. (2020a) showed that contextualised embeddings such as BERT (Devlin et al., 2019) and ELMo (Peters et al., 2018a) can be used to create sense embeddings by means of external semantic networks, such as WordNet (Miller, 1998) and BabelNet (Navigli and Ponzetto, 2010). Moreover, Levine et al. (2020) showed that BERT can be fine-tuned using WordNet's supersenses to learn contextualised sense embeddings.

Inspired by these prior successes, we ask and affirmatively answer the question – *can we extract sense-related information from contextualised word embeddings to create sense-specific versions of (pretrained) sense-agnostic static embeddings?* To this end, we propose, Context Derived Embeddings of Senses (**CDES**) a method to extract sense-related information encoded in contextualised word embeddings and inject it into pretrained sense-agnostic static word embeddings to create sense-specific static embeddings. Given a contextualised embedding, a static word embedding and a sense-annotated corpus, CDES learns sense-specific projection matrices that can be used to predict the sense embeddings of words from their word embeddings. Following the distributional hypothesis (Harris, 1954), we require that the predicted sense embedding of a word must align (possibly non-linearly) with the meaning of the context of the word, represented using a contextualised embedding as outlined in Figure 1.

At a more conceptual level, CDES can be seen as using contextualised language models as a proxy for extracting information relevant to a particular task, without learning it directly from text corpora. In particular, prior work probing language models has shown that rich information about languages is compactly and accurately encoded within contextualised representations produced by NLMs (Klafka and Ettinger, 2020). Moreover, CDES can also be seen as an instance of *model distillation* (Furlanello et al., 2018), where a complex teacher model (i.e. a contextualsied word embedding) is used to train a simpler student model (i.e. a sense-sensitive static embedding).

There are several advantages in CDES for learning sense-specific static embeddings. CDES is computationally relatively lightweight because it uses *pretrained* static embeddings as well as contextualised embeddings from a *pretrained* NLM and does not require training these resources from scratch. CDES static sense embeddings can be precomputed because of their independence on the context. Therefore, CDES embeddings are attractive to NLP applications that must run on limited hardware resources. Because subtokenisation methods, such as Byte Pair Encoding (BPE), must be used to limit the vocabulary sizes, one must post-process subtoken embeddings (e.g. by mean pooling) to create word embeddings with contextualised embeddings, whereas static embeddings can directly learn word embeddings. To increase the coverage of sense embeddings, in addition to the sense related information extracted from contextualised embeddings, CDES incorporates contextual information from external corpora and knowledge bases.

We evaluate CDES on Word Sense Disambiguation (Navigli, 2009, WSD) (Section 3.2) and Words in Context (Pilehvar and Camacho-Collados, 2019, WiC) (Section 3.3) tasks. In both tasks, CDES learns accurate sense embeddings and outperforms many existing static sense embeddings. In particular, on the WSD framework (Raganato et al., 2017), CDES reports the best performance in 4 out of 6 benchmarks, and on WiC reports competitive results to the current state-of-the-art without any fine-tuning of on task data.

2 Context-Derived Embedding of Senses

Given (a) pretrained static word embeddings, (b) contextualised word embeddings from a pretrained NLM, and (c) a sense-annotated corpus, CDES learns a sense-specific version of (a), representing each sense of a word by a different vector. To describe CDES in detail, let us denote the sense-agnostic static embedding of a word $u \in \mathcal{V}$ in a vocabulary $\mathcal{V}$, by $\boldsymbol{g}(u) \in \mathbb{R}^p$. Moreover, let us denote the contextualised embedding model c, from which we can obtain a context-sensitive representation $\boldsymbol{c}(u, t) \in \mathbb{R}^q$ corresponding to u in some context $t \in \mathcal{C}(u)$. Here, $\mathcal{C}(u)$ is the set of contexts in which u occurs. An

ambiguous word u is likely to take different senses in different contexts t, and our goal is to learn a sense-specific embedding of u that captures the different senses of u.

Let us denote by $\mathcal{S}$ the set of word senses taken by all words in $\mathcal{V}$. An ambiguous word u will belong to a subset $\mathcal{S}(u)$ of senses in $\mathcal{S}$. Let us denote the sense-specific embedding of u corresponding to the i-th sense $s_i \in \mathcal{S}(u)$ by $\boldsymbol{s}_i(u) \in \mathbb{R}^p$. We model the process of creating sense-specific embeddings from static embeddings as a projection learning task, where we multiply the static embedding, $\boldsymbol{g}(u)$, by a sense-specific projection matrix, $\mathbf{A}_i$, to produce $\boldsymbol{s}_i(u)$ as in (1).

$$\boldsymbol{s}_i(u) = \mathbf{A}_i \boldsymbol{g}(u) \tag{1}$$

Here, (1) decouples a sense embedding into a sense-agnostic static lexical semantic component given by $\boldsymbol{g}(u)$ and a word-independent sense-specific component $\mathbf{A}_i$, enabling efficient sense-specific embedding learning using pretrained embeddings. The projection matrices can be seen as linear operators that produce different sense-specific embeddings from the same static word (lemma) embedding, corresponding to the different senses of the lemma.

On the other hand $\boldsymbol{c}(u, t)$ encodes both sense related information for u as well as information not related to u such as the grammatical gender or number in the context t. Therefore, we apply a linear filter parameterised by a matrix $\mathbf{W} \in \mathbb{R}^{q \times p}$, to extract sense related information from $\boldsymbol{c}(u, t)$.

Given a sense tagged corpus, we jointly learn $\mathbf{W}$ and $\mathbf{A}_i$s by minimising the objective given by (2).

$$L(\mathbf{W}, \{\mathbf{A}_i\}_{i=1}^{|\mathcal{S}|}) = \sum_{\substack{u \in \mathcal{V} \\ t \in \mathcal{C}(u) \\ s_i \in \mathcal{S}(u)}} ||\mathbf{W}\boldsymbol{c}(u, t) - f(\mathbf{A}_i \boldsymbol{g}(u))||_2^2$$
$$\tag{2}$$

Here, f is an elementwise nonlinear function that enables us to consider nonlinear associations between contextualised and static word embeddings. In our experiments, we consider linear, ReLU and GELU activations as f. After training, we can compute the sense embeddings $\boldsymbol{s}_i(u)$ using (1) with the pretrained static word embeddings $\boldsymbol{g}(u)$.

Eq. (2) can be seen as aligning the contextualised and static word embeddings under a nonlinear transformation. The only learnable parameters in our proposed method are $\mathbf{W}$ and sense-specific projections $\mathbf{A}_1, \ldots, \mathbf{A}_{|\mathcal{S}|}$. In particular, we *do not* require re-training or fine-tuning static or contextualised embeddings and can be seen as a post-processing method applied to pretrained embeddings, similar to retrofitting (Shi et al., 2019). We limit the sense-specific projection matrices to diagonal matrices in our experiments because in our preliminary investigations we did not find any significant advantage in using full matrices compared to the extra storage. Moreover, a diagonal matrix can be compactly represented by storing only its diagonal elements as a vector, which reduces the number of parameters to learn (thus less likely to overfit) and speeds up matrix-vector multiplications.

2.1 Context Aggregation

An important limitation of the above-mentioned setting is that it requires sense-annotated corpora. Manually annotating word senses in large text corpora is expensive and time consuming. Moreover, such resources might not be available for low resource languages. Even if such sense-annotated corpora are available for a particular language, they might not cover all different senses of all of the words in that language, resulting in an inadequate sense coverage. For example, SemCor (Miller et al., 1993), one of the largest manually-annotated corpora for English word senses including more than 220K words tagged with 25K distinct WordNet meanings, covers only 15% of all synsets in the WordNet. To address this sense-coverage problem, we follow prior proposals (Scarlini et al., 2020b) to extract additional contexts for a word from (a) **the dictionary definitions of synsets**, and (b) **an external corpus**.

Gloss-based Sense Embeddings: To create sense embeddings from dictionary definitions, we use the glosses of synsets in the WordNet. Given a word u, we create a gloss-based sense embedding, $\boldsymbol{\psi}(u)_i \in \mathbb{R}^q$, represented by the sentence embedding, $\boldsymbol{c}(t_i)$, computed from the gloss t_i corresponding to the synset s_i of u. Here, $\boldsymbol{c}(t_i)$ is computed by averaging the contextualised embeddings for the tokens in the gloss t_i as given in (3).

$$\boldsymbol{c}(t_i) = \operatorname*{avg}_{w \in t_i} \boldsymbol{c}(w, t_i) \tag{3}$$

Here, avg denotes mean pooling over the tokens w in t_i. Following Loureiro and Jorge (2019a) and Scarlini et al. (2020b), in our experiments, we use BERT as the contextualised embedding model and use the sum of the final four layers as token embeddings.

Corpus-based Sense Embeddings: To extract contexts from an external corpus for given a word u, we retrieve all sentences as contexts $t \in \mathcal{C}(u)$ from the corpus where u occurs. We then cluster the extracted sentences (represented by the sentence embeddings computed using (3)) using the k-means algorithm. We assume each cluster to contain similar sentences and that u will be used in the same sense in all sentences in a cluster. We use UKB[1] (Agirre et al., 2014), a knowledge-based approach to WSD that uses the Personalised PageRank algorithm (Haveliwala et al., 2002), to disambiguate the clusters.

To increase the coverage of senses represented by the clusters, we consider collocations of u available in SyntagNet (Maru et al., 2019)[2] following Scarlini et al. (2020b). Specifically, for each word u, we find words v that forms a collocation with u in SyntagNet, and extract sentences t that contain both u and v within a co-occurrence window. The synset id s_i assigned to the (u, v) pair in SyntagNet is used as the sense id for all extracted sentences for u. Finally, we compute a corpus-based sense embedding $\phi_i(u) \in \mathbb{R}^q$ as the cluster centroid, where sentence embeddings are computed using (3).

2.2 Sense Embedding and Disambiguation

The final CDES static sense embedding, $\mathbf{cdes}_i(u) \in \mathbb{R}^{p+2q}$ of the i-th sense of u is computed as the concatenation of $\mathbf{s}_i(u)$ (given by (1)), gloss-based sense embedding $\psi_i(u)$ and corpus-based sense embedding $\phi_i(u)$ as given by (4), where $\oplus$ denotes vector concatenation.

$$\mathbf{cdes}_i(u) = \mathbf{s}_i(u) \oplus \psi_i(u) \oplus \phi_i(u) \qquad (4)$$

In order to disambiguate a word u in a given context t', we first compute a contextualised embedding $\zeta(u, t') \in \mathbb{R}^{p+2q}$ by concatenating three vectors as give by (5)

$$\zeta(u, t') = \mathbf{g}(u) \oplus \mathbf{c}(u, t') \oplus \mathbf{c}(u, t') \qquad (5)$$

We then compute the cosine similarity between $\zeta(u, t')$ and $\mathbf{cdes}_i(u)$ for each sense s_i of u. We limit the candidate senses based on the lemma and part-of-speech of u in t', and select the most similar (1-NN) sense of u as its disambiguated sense in context t'.

3 Experiments

3.1 Experimental Setup

In our experiments, we use the pretained GloVe[3] embeddings (Common Crawl with 840B tokens and 2.2M vocabulary) as the static word embeddings $\mathbf{g}(u)$ with $p = 300$. We use pretrained BERT (`large-bert-cased`[4]) as the contextualised embedding model, $\mathbf{c}(u, t)$ with $q = 1024$. Following prior work (Luo et al., 2018a; Luo et al., 2018b; Loureiro and Jorge, 2019a; Scarlini et al., 2020b), we use sense annotations from SemCor 3.0 (Miller et al., 1993) as the sense-tagged corpus, which is the largest corpus annotated with WordNet sense ids. As the external corpus for extracting contexts as described in Section 2.1, we use the English Wikpedia. The number of clusters in k-means is set to the number of distinct senses for the lexeme according to the WordNet. The number of words given to UKB is set to 5 and the number of sentences extracted from Wikipedia per lemma is set to 150 following Scarlini et al. (2020b). The co-occurrence window size for considering collocations extracted from SyntagNet is set to 3 according to Maru et al. (2019). We evaluate the learnt sense embeddings in two downstream tasks: WSD (Section 3.2) and WiC (Section 3.3). The statistics of SemCor, all-words English WSD and WiC datasets are showed in Table 1.

To project contextualised and static word embeddings to a common space, we set $\mathbf{W} \in \mathbb{R}^{300 \times 1024}$. To reduce the memory footprint, number of trainable parameters and thereby overfitting, we constrain the sense-specific matrices $\mathbf{A}_i \in \mathbb{R}^{300 \times 300}$ to be diagonal. We initialise all elements of $\mathbf{W}$ and $\mathbf{A}_i$s uniformly at random in $[0, 1]$. We use Adam as the SGD optimiser and set the minibatch size to 64 with an initial learning rate of 1E-4. All hyperparameter values were tuned using a randomly selected subset of training data set aside as a valida-

---

[1] http://ixa2.si.ehu.eus/ukb/
[2] http://syntagnet.org/

[3] nlp.stanford.edu/projects/glove/
[4] https://bit.ly/33Nsmou

Dataset	Total	Nouns	Vebs	Adj	Adv
SemCor	226,036	87,002	88,334	31,753	18,947
WSD					
SE2	2,282	1,066	517	445	254
SE3	1,850	900	588	350	12
SE07	455	159	296	-	-
SE13	1,644	1,644	-	-	-
SE15	1,022	531	251	160	80
ALL	7,253	4,300	1,652	955	346

WiC	Instances	Nouns	Vebs	Unique Words	
Training	5,428	2,660	2,768	1,256	
Dev	638	396	242	599	
Test	1,400	826	574	1,184	

Table 1: The statistics of the training and evaluation datasets. SemCor is used for training. SemEval (SE07, SE13, SE15) and Senseval (SE2, SE3) datasets are used for the WSD task, whereas the WiC dataset is used for sense discrimination task.

tion dataset. The t-SNE visualisations in the paper are produced with `sklearn.manifold.TSNE` using n_components=2, init=*pca*, perplexity=3, n_iter=1500 and metric=*cosine*.

All experiments were conducted on a machine with a single Titan V GPU (12 GB RAM), Intel Xeon 2.60 GHz CPU (16 cores) and 64 GB of RAM. Overall, training time is less than 3 days on this machine,

3.2 Word Sense Disambiguation (WSD)

WSD is a fundamental task in NLP, which aims to identify the exact sense of an ambiguous word in a given context (Navigli, 2009). To evaluate the proposed sense embeddings, we conduct a WSD task using the evaluation framework proposed by Raganato et al. (2017), which includes all-words English WSD datasets: Senseval-2 (SE2), Senseval-3 task 1 (SE3), SemEval-07 task 17 (SE07), SemEval-13 task 12 (SE13) and SemEval-15 task 13 (SE15). We used the framework's official scoring scripts to avoid any discrepancies in the scoring methodology. As described in Section 2.2, the sense of a word in a context is predicted by the 1-NN method.

Table 2 shows the WSD results. Most Frequent Sense (MFS) baseline selects the most frequent sense of a word in the training corpus and has proven to be a strong baseline (McCarthy et al., 2007). Scarlini et al. (2020b) use Peters et al. (2018a)'s method with BERT on SemCor+OMSTI (Taghipour and Ng, 2015) to propose SemCor+OMSTI$_{BERT}$ baseline.

ELMo k-NN uses ELMo embeddings to predict the sense of a word following the nearest neighbour strategy. Specifically, they first obtain ELMo embeddings for all words in SemCor sentences, and average the embeddings for each sense. At test time, they run ELMo on the given test sentence containing the ambiguous word and select the sense with the highest cosine similarity. Loureiro and Jorge (2019a) repeated this method using BERT (Devlin et al., 2019) embeddings to propose the BERT k-NN baseline. EWISE$_{ConvE}$ (Kumar et al., 2019) learns a sentence encoder for sense definition by using WordNet relations as well as ConvE (Dettmers et al., 2018). Scarlini et al. (2020b) report the performance of using BERT base-multilingual-cased (mBERT) instead of BERT large with MFS fallback. Hadiwinoto et al. (2019) integrating pretrained BERT model with gated linear unit (GLU) and layer weighted (LW).

GlossBERT (Huang et al., 2019) fine tunes the pretrained BERT model by jointly encoding contexts and glosses. LMMS (Loureiro and Jorge, 2019a) learns sense embeddings using BERT to generate contextualised embeddings from semantic networks and sense definitions. To perform WSD, they use the 1-NN method and compare sense embeddings against contextualised embeddings generated by BERT. Scarlini et al. (2020b) augment UKB with SyntagNet's relations (Scozzafava et al., 2020) and obtain UKB$_{+Syn}$. SensEmBERT is a knowledge-based approach, which produces sense embeddings by means of BabelNet and Wikipedia. Although SensEmBERT is effective in modelling nominal meanings, it only consists of nouns due to the limitation of its underlying resources. SensEmBERT$_{sup}$ is the supervised version of SensEmBERT. ARES (Scarlini et al., 2020b) is a semi-supervised approach for learning sense embeddings by incorporating sense annotated datasets, unlabelled corpora and knowledge bases.

To study the effect of using a nonlinear mapping f between static and contextualised embedding spaces in (2), we train CDES with linear, ReLU and GELU activations to create respectively CDES$_{linear}$, CDES$_{ReLU}$ and CDES$_{GELU}$ versions. From Table 2 we see that among these versions, CDES$_{GELU}$ outperforms the linear and ReLU versions in all datasets, except on SE2 where CDES$_{linear}$ performs best. This result shows that nonlinear mapping (GELU) to be more appropriate for extracting sense-related

Models	SE2	SE3	SE07	SE13	SE15	ALL
MFS	65.6	66.0	54.5	63.8	67.1	65.6
SemCor+OMSTI$_{BERT}$	74.0	70.6	63.1	72.4	75.0	72.2
ELMo k-NN	71.5	67.5	57.1	65.3	69.9	67.9
BERT k-NN	76.3	73.2	66.2	71.7	74.1	73.5
EWISE$_{ConvE}$	73.8	71.1	67.3	69.4	74.5	71.8
mBERT k-NN + MFS	72.7	70.1	62.4	69.0	72.0	70.5
BERT$_{GLU+LW}$	75.5	73.4	68.5	71.0	76.2	74.0
GlossBERT	77.7	75.2	**76.1**	72.5	80.4	77.0
LMMS	76.3	75.6	68.1	75.1	77.0	75.4
UKB$_{+Syn}$	71.2	71.6	59.6	72.4	75.6	71.5
SensEmBERT	70.8	65.4	58.0	74.8	75.0	70.1
SenseEmBERT$_{sup}$	72.2	69.9	60.2	**78.7**	75.0	72.8
ARES	78.0	<u>77.1</u>	71.0	77.3	**83.2**	77.9
Proposed Method						
CDES$_{linear}$	**78.4**	76.9	71.0	77.6	<u>83.1</u>	<u>78.0</u>
CDES$_{ReLU}$	<u>78.1</u>	<u>77.1</u>	71.0	77.5	<u>83.1</u>	<u>78.0</u>
CDES$_{GELU}$	<u>78.1</u>	**77.3**	<u>71.4</u>	<u>77.7</u>	**83.2**	**78.1**

Table 2: F1 scores (%) for English all-words WSD on the test sets of Raganato et al. (2017). Bold and underline indicate the best and the second best results, respectively.

Models	Accuracy %
Static Embeddings	
GloVe (Pennington et al., 2014)	50.9
Contextualised Embeddings	
ElMo (Peters et al., 2018b)	57.7
ELMo-weighted (Ansell et al., 2019)	61.2
BERT-large (Devlin et al., 2019)	65.5
RoBERTa (Liu et al., 2019)	69.9
KnowBERT-W+W (Peters et al., 2019)	70.9
SenseBERT-large (Levine et al., 2020)	<u>72.1</u>
BERT$_{ARES}$ (Scarlini et al., 2020b)	**72.2**
Static Sense Embeddings	
MUSE (Lee and Chen, 2017)	48.4
LMMS (Loureiro and Jorge, 2019b)	67.7
LessLex (Colla et al., 2020)	59.2
CDES$_{linear}$	69.0
CDES$_{ReLu}$	68.6
CDES$_{GELU}$	68.8

Table 3: Performance on WiC. Bold and underline respectively indicate the best and the second best results.

information from contextualised embeddings. Moreover, we see that CDES versions consistently outperform all previously proposed sense embeddings, except on SE07 and SE13 where GlossBERT and SenseBERT$_{sup}$ performs best respectively. On SE15, the performance of CDES$_{GELU}$ is equal to that of ARES.

Overall, CDES$_{linear}$ obtains the best performance on SE2, while CDES$_{GELU}$ performs best on SE3, SE15 and ALL. This result provides empirical support to our working hypothesis that contextualised embeddings produced by NLMs encode much more information beyond sense related information, which must be filtered out using **W**. CDES is able to accurately extract the sense-specific information from contextualised embeddings generated by a pretrained NLM to create sense-specific versions of pretrained sense-agnostic static embeddings.

3.3 Words in Context (WiC)

Pilehvar and Camacho-Collados (2019) introduced the WiC dataset for evaluating sense embedding methods. For a particular word u, WiC contains pairs of sentences, (t_1, t_2) where the same (*positive*) or different (*negative*) senses of u can occur. An accurate sense embedding method must be able to discriminate the different senses of an ambiguous word. The problem is formalised as a binary classification task and classification accuracy is reported

as the evaluation metric. A method that assigns the same vector to all of the senses of a word would report a chance-level (i.e. 50%) accuracy on WiC.

Similar to Section 3.2, we first determine the sense-specific embeddings of u, $s_i(u)$ and $s_j(u)$ for the senses of u used in respectively t_1 and t_2. We then train a binary logistic regression classifier using the train split of WiC, where we use the cosine similarities between the two vectors in the following six pairs as features, comparing sense and contextualised embeddings in the two sentences.: $(s_i(u), s_j(u))$, $(\zeta(u, t_1), \zeta(u, t_2))$, $(s_i(u), \zeta(u, t_1))$, $(s_j(u), \zeta(u, t_2))$, $(s_i(u), \zeta(u, t_2))$ and $(s_j(u), \zeta(u, t_1))$. We train this classifier using the official train split in WiC. In particular, we do not fine-tune the static or contextualised embeddings that are used as inputs by CDES on WiC because our goal is to extract sense-related information already present in the pretrained embeddings.

In Table 3, we report the classification accuracies on WiC for different types of embeddings such as static word embeddings (GloVe), contextualised embeddings generated by NLMs (ELMo, ElMo-weighted, BERT-large, RoBERTa and KnowBERT), and sense-specific embeddings (MUSE, LMMS, LessLex, SenseBERT-large and BERT$_{ARES}$). Due
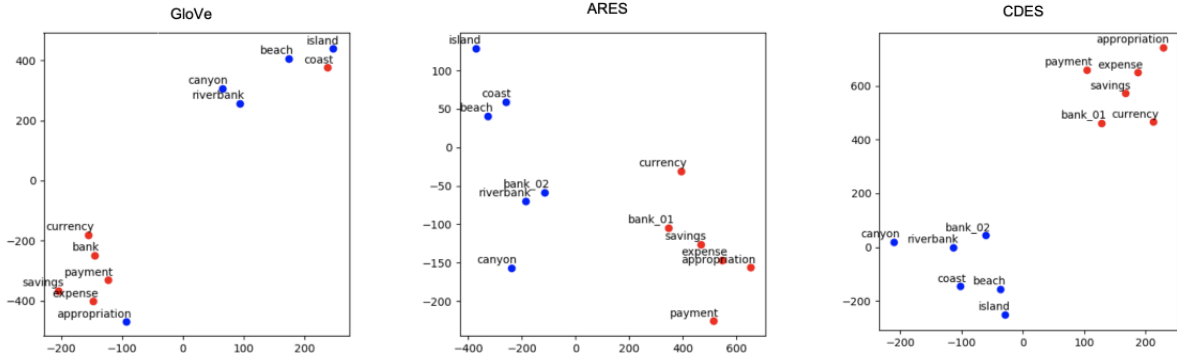
Figure 2: t-SNE visualisations of the nearest neighbours of *bank* corresponding to the two senses *financial institution* (in red) and *sloping land* (in blue) are shown for GloVe, ARES and CDES embeddings. Sense labels of synonyms are omitted to avoid cluttering.

to space limitations we omit the details of these embeddings.

From Table 3 we see that SenseBERT-large and BERT_{ARES} obtain better performance than other embeddings. All the CDES variants outperform previous static sense embeddings learning methods. However, MUSE[5] do not assign sense labels to sense embeddings as done by LMMS, LessLex and CDES. Among CDES variants, CDES_{linear} performs best and is closely followed by GELU and ReLU variants. Although, CDES variants do not surpass the current SoTA methods such as SenseBERT-large and BERT_{ARES} on WiC, unlike CDES these methods fine-tune on WiC train data and/or use more complex classifiers with multiple projection layers compared to the single logistic regression over six features used by CDES.[6] More importantly, results from both WSD and WiC experiments support our claim that contextualised embeddings encode word sense related information that can be extracted and injected into sense-insensitive static word embeddings via (non)linear projections to create sense-sensitive versions of the sense-insensitive static embeddings.

3.4 Visualisation of Sense Embeddings

To visualise the embeddings corresponding to the different senses of an ambiguous word, we consider *bank*, which has the two distinct senses: *financial institution* and *sloping land*. We randomly select 5

synonyms for each sense of *bank* from the Word-Net and project their sense/word embeddings using t-SNE in Figure 2. Compared to GloVe, we see that words with related meanings are projected onto coherent clusters by ARES and CDES. This indicates that sense embeddings are able to distinguish polysemy correctly compared to static word embeddings. Overall, we see that CDES produces better separated clusters than both GloVe and ARES.

3.5 Nearest Neighbours of Sense Embeddings

An accurate sense embedding method must be able to represent an ambiguous word with different embeddings considering the senses expressed by that word in different contexts. To understand how sense embedding of a word vary in different contexts, we compute the nearest neighbours of an ambiguous word using its sense embedding. Table 4 shows two sentences from SemCor containing *bank*, where in Sentence 1, *bank* takes the *financial institution* sense, and in Sentence 2 the *sloping land (especially the slope beside a body of water)* sense. We compute the sense embedding of *bank*, given each sentence as the context, using different methods and compute the top 5 nearest neighbours, shown in the descending order of their cosine similarity scores with the sense embedding of *bank* in each sentence.

GloVe, which is sense and context insensitive uses the same vector to represent *bank* in both sentences, resulting in the same set of nearest neighbours, which is a mixture of finance and riverbank related words. On the other hand, BERT, which is context-sensitive

Sentence 1: The <mark>banks</mark> which held the mortgage on the old church declared that the interest was considerably in arrears, and the real estate people said flatly that the land across the river was being held for an eventual development for white working people who were coming in, and that none would be sold to colored folk.				
GloVe	BERT	LMMS	SenseBERT	CDES
mortgage	mortgage	mortgage	mortgage	mortgage
interest	interest	church	real	real estate
estate	held	sell	old	sell
river	church	interest	land	interest
real	river	real estate	interest	church
Sentence 2: Through the splash of the rising waters, they could hear the roar of the river as it raged through its canyon, gnashing big chunks out of the <mark>banks</mark>.				
GloVe	BERT	LMMS	SenseBERT	CDES
mortgage	river	river	splash	river
interest	waters	canyon	land	water
estate	chunks	land	out	rise
river	splash	folk	through	canyon
real	canyon	church	chunks	folk

Table 4: Nearest neighbours computed using the word/sense embeddings of *bank* in two sentences.

but not sense-specific, returns different sets of nearest neighbours in the two cases. In particular, we see that finance-related nearest neighbours such as *mortgage* and *interest* are selected for the first sentence, whereas riverbank-related nearest neighbours such as *water* and *canyon* for the second. However, BERT does not provide sense embeddings and some neighbours such as *river* appear in both sets, because it appears in the first sentence, although not related to *bank* there.

SenseBERT (Levine et al., 2020) disambiguates word senses at a coarse-grained WordNet's supersense level. We see that SenseBERT correctly detects words such as *mortgage* and *interest* as neighbours of *bank* in the first sentence, and *splash* and *land* in the second. We see that *land* appears as a nearest neighbour in both sentences, although it is more related to the *sloping land* sense than the *financial institution* sense of *bank*.

LMMS selects *church* as a nearest neighbour for both sentences, despite being irrelevant to the second. On the other hand, CDES correctly detects *church* for the first sentence and not for the second. Overall, CDES correctly lists financial institution sense related words such as *mortgage*, *real estate* and *interest* for the first sentence, and sloping land sense related words such as *river*, *water* and *canyon* in the

second sentence.

4 Related Work

Reisinger and Mooney (2010) proposed multi-prototype embeddings to represent word senses, which was extended by Huang et al. (2012) combining both local and global contexts. Both methods use clustering to group contexts of a word related to the same sense. Although the number of senses depends on the word, these methods assign a fixed number of senses to all words. To overcome this limitation, Neelakantan et al. (2014) proposed a non-parametric model, which estimates the number of senses dynamically per word.

Even though clustering-based methods are able to assign multi-prototype embeddings for a word, they still suffer from the fact that the trained embeddings are not associated with any sense inventories (Camacho-Collados and Pilehvar, 2018). In contrast, knowledge-based approaches learn sense embeddings by extracting sense-specific information from external sense inventories, such as Word-Net and BabelNet. Chen et al. (2014) extended word2vec (Mikolov et al., 2013) to learn sense-specific embeddings associated with the Word-Net (Miller, 1998) *synsets*. Rothe and Schütze (2015) used the semantic relations in WordNet to embed

words and their senses into a common vector space. Iacobacci et al. (2015) use the sense definitions from BabelNet and perform word sense disambiguation (WSD) to obtain sense-specific contexts.

Recently, contextualised embeddings generated by NLMs have been used to create sense embeddings. Loureiro and Jorge (2019a) construct sense embeddings by taking the average over the contextualised embeddings of the sense annotated tokens from Sem-Cor. Scarlini et al. (2020a) propose a knowledge-based approach for constructing BERT-based embeddings of senses by means of the lexical-semantic information in BabelNet and Wikipedia. CDES proposed in this paper extends this line of work, where we extract sense-related information from an NLM and inject into a static word embedding to create a sense-specific version of the latter. Moreover, we follow prior work (Scarlini et al., 2020b; Loureiro and Jorge, 2019a) and incorporate contexts from external resources such as Wikipedia, WordNet and Syntag-Net representing different senses of a word to enhance sense embeddings learnt using sense-labelled corpora.

5 Conclusion

We proposed CDES, a method which is able to generate sense embeddings by extracting the sense-related information from contextualised embeddings. CDES integrates the gloss information from a semantic network as well as the information from an external corpus to tackle the sense-coverage problem. Evaluations on multiple benchmark datasets related to WSD and WiC tasks show that CDES learns accurate sense emebddings, and report comparable results to the current SoTA. All experiments reported in the paper are limited to English language and we plan to extend the proposed method to learn multilingual sense embeddings in our future work.

References

[Agirre et al.2014] Eneko Agirre, Oier López de Lacalle, and Aitor Soroa. 2014. Random walks for knowledge-based word sense disambiguation. *Computational Linguistics*, 40(1):57–84.

[Ansell et al.2019] Alan Ansell, Felipe Bravo-Marquez, and Bernhard Pfahringer. 2019. An elmo-inspired approach to semdeep-5's word-in-context task. In *SemDeep-5*, pages 21–25.

[Camacho-Collados and Pilehvar2018] Jose Camacho-Collados and Mohammad Taher Pilehvar. 2018. From word to sense embeddings: A survey on vector representations of meaning. *Journal of Artificial Intelligence Research*, 63:743–788.

[Chen et al.2014] Xinxiong Chen, Zhiyuan Liu, and Maosong Sun. 2014. A unified model for word sense representation and disambiguation. In *EMNLP*, pages 1025–1035.

[Colla et al.2020] Davide Colla, Enrico Mensa, and Daniele P. Radicioni. 2020. LessLex: Linking multilingual embeddings to SenSe representations of LEXical items. *Computational Linguistics*, 46(2):289–333.

[Dettmers et al.2018] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In *AAAI*, volume 32.

[Devlin et al.2019] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, pages 4171–4186.

[Furlanello et al.2018] Tommaso Furlanello, Zachary Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. 2018. Born-again neural networks. In *ICML*, volume 80 of *Machine Learning Research*, pages 1602–1611.

[Hadiwinoto et al.2019] Christian Hadiwinoto, Hwee Tou Ng, and Wee Chung Gan. 2019. Improved word sense disambiguation using pre-trained contextualized word representations. In *EMNLP-IJCNLP*, pages 5300–5309.

[Harris1954] Zellig Harris. 1954. Distributional structure. *Word*, 10(23):146–162.

[Haveliwala et al.2002] Taher H. Haveliwala, Aristides Gionis, Dan Klein, and Piotr Indyk. 2002. Evaluating strategies for similarity search on the web. In *WWW*.

[Huang et al.2012] Eric H Huang, Richard Socher, Christopher D Manning, and Andrew Y Ng. 2012. Improving word representations via global context and multiple word prototypes. In *ACL*, pages 873–882.

[Huang et al.2019] Luyao Huang, Chi Sun, Xipeng Qiu, and Xuan-Jing Huang. 2019. Glossbert: Bert for word sense disambiguation with gloss knowledge. In *EMNLP-IJCNLP*, pages 3500–3505.

[Iacobacci et al.2015] Ignacio Iacobacci, Mohammad Taher Pilehvar, and Roberto Navigli. 2015. Sensembed: Learning sense embeddings for word and relational similarity. In *ACL-IJCNLP*, pages 95–105.

[Klafka and Ettinger2020] Josef Klafka and Allyson Ettinger. 2020. Spying on your neighbors: Fine-grained

probing of contextual embeddings for information about surrounding words. In *ACL*, pages 4801–4811.

[Kumar et al.2019] Sawan Kumar, Sharmistha Jat, Karan Saxena, and Partha Talukdar. 2019. Zero-shot word sense disambiguation using sense definition embeddings. In *ACL*, pages 5670–5681.

[Lee and Chen2017] Guang-He Lee and Yun-Nung Chen. 2017. Muse: Modularizing unsupervised sense embeddings. In *EMNLP*, pages 327–337.

[Levine et al.2020] Yoav Levine, Barak Lenz, Or Dagan, Ori Ram, Dan Padnos, Or Sharir, Shai Shalev-Shwartz, Amnon Shashua, and Yoav Shoham. 2020. Sense-BERT: Driving some sense into BERT. In *ACL*, pages 4656–4667.

[Liu et al.2019] 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. *arXiv preprint arXiv:1907.11692*.

[Loureiro and Jorge2019a] Daniel Loureiro and Alipio Jorge. 2019a. Language modelling makes sense: Propagating representations through wordnet for full-coverage word sense disambiguation. In *ACL*, pages 5682–5691.

[Loureiro and Jorge2019b] Daniel Loureiro and Alípio Mário Jorge. 2019b. Liaad at semdeep-5 challenge: Word-in-context (wic). In *SemDeep@IJCAI*.

[Luo et al.2018a] Fuli Luo, Tianyu Liu, Zexue He, Qiaolin Xia, Zhifang Sui, and Baobao Chang. 2018a. Leveraging gloss knowledge in neural word sense disambiguation by hierarchical co-attention. In *EMNLP*, pages 1402–1411.

[Luo et al.2018b] Fuli Luo, Tianyu Liu, Qiaolin Xia, Baobao Chang, and Zhifang Sui. 2018b. Incorporating glosses into neural word sense disambiguation. In *ACL)*, pages 2473–2482.

[Maru et al.2019] Marco Maru, Federico Scozzafava, Federico Martelli, and Roberto Navigli. 2019. SyntagNet: Challenging supervised word sense disambiguation with lexical-semantic combinations. In *EMNLP-IJCNLP*, pages 3534–3540.

[McCarthy et al.2007] Diana McCarthy, Rob Koeling, Julie Weeds, and John Carroll. 2007. Unsupervised acquisition of predominant word senses. *Computational Linguistics*, 33(4):553 – 590.

[Mikolov et al.2013] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS*, pages 3111–3119.

[Miller et al.1993] George A. Miller, Claudia Leacock, Randee Tengi, and Ross T. Bunker. 1993. A semantic concordance. In *Human Language Technology: In a Workshop Held at Plainsboro, 1993*.

[Miller1998] George A Miller. 1998. *WordNet: An electronic lexical database*.

[Navigli and Ponzetto2010] Roberto Navigli and Simone Paolo Ponzetto. 2010. Babelnet: Building a very large multilingual semantic network. In *ACL*, pages 216–225.

[Navigli2009] Roberto Navigli. 2009. Word sense disambiguation: A survey. *ACM computing surveys (CSUR)*, 41(2):1–69.

[Neelakantan et al.2014] Arvind Neelakantan, Jeevan Shankar, Alexandre Passos, and Andrew McCallum. 2014. Efficient non-parametric estimation of multiple embeddings per word in vector space. In *EMNLP*, pages 1059–1069.

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

[Peters et al.2018a] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018a. Deep contextualized word representations. In *NAACL-HLT*.

[Peters et al.2018b] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018b. Deep contextualized word representations. In *NAACL-HLT*, pages 2227–2237.

[Peters et al.2019] Matthew E Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A Smith. 2019. Knowledge enhanced contextual word representations. In *EMNLP-IJCNLP*, pages 43–54.

[Pilehvar and Camacho-Collados2019] Mohammad Taher Pilehvar and Jose Camacho-Collados. 2019. WiC: the word-in-context dataset for evaluating context-sensitive meaning representations. In *NAACL*, pages 1267–1273.

[Raganato et al.2017] Alessandro Raganato, Jose Camacho-Collados, and Roberto Navigli. 2017. Word sense disambiguation: A unified evaluation framework and empirical comparison. In *EACL*, pages 99–110.

[Reisinger and Mooney2010] Joseph Reisinger and Raymond Mooney. 2010. Multi-prototype vector-space models of word meaning. In *NAACL-HLT*, pages 109–117.

[Rothe and Schütze2015] Sascha Rothe and Hinrich Schütze. 2015. Autoextend: Extending word embeddings to embeddings for synsets and lexemes. In *ACL-IJCNLP*, pages 1793–1803.

[Scarlini et al.2020a] Bianca Scarlini, Tommaso Pasini, and Roberto Navigli. 2020a. Sensembert: Context-enhanced sense embeddings for multilingual word sense disambiguation. In *AAAI*, number 05, pages 8758–8765.

[Scarlini et al.2020b] Bianca Scarlini, Tommaso Pasini, and Roberto Navigli. 2020b. With more contexts comes better performance: Contextualized sense embeddings for all-round word sense disambiguation. In *EMNLP*, pages 3528–3539.

[Scozzafava et al.2020] Federico Scozzafava, Marco Maru, Fabrizio Brignone, Giovanni Torrisi, and Roberto Navigli. 2020. Personalized pagerank with syntagmatic information for multilingual word sense disambiguation. In *ACL*, pages 37–46.

[Shi et al.2019] Weijia Shi, Muhao Chen, Pei Zhou, and Kai-Wei Chang. 2019. Retrofitting contextualized word embeddings with paraphrases. In *EMNLP-IJCNLP*, pages 1198–1203.

[Taghipour and Ng2015] Kaveh Taghipour and Hwee Tou Ng. 2015. One million sense-tagged instances for word sense disambiguation and induction. In *CoNLL*, pages 338–344.