Together We Make Sense– Learning Meta-Sense Embeddings from Pretrained Static Sense Embeddings

Haochen Luo[†] **Yi Zhou**[♢] **Danushka Bollegala**[†,‡]
University of Liverpool[†], Cardiff University[♢], Amazon[‡]
haochen.luo@outlook.com
danushka@liverpool.ac.uk
zhouy131@cardiff.ac.uk

Abstract

Sense embedding learning methods learn multiple vectors for a given ambiguous word, corresponding to its different word senses. For this purpose, different methods have been proposed in prior work on sense embedding learning that use different sense inventories, sense-tagged corpora and learning methods. However, not all existing sense embeddings cover all senses of ambiguous words equally well due to the discrepancies in their training resources. To address this problem, we propose the first-ever meta-sense embedding method – Neighbour Preserving Meta-Sense Embeddings, which learns meta-sense embeddings by combining multiple independently trained source sense embeddings such that the sense neighbourhoods computed from the source embeddings are preserved in the meta-embedding space. Our proposed method can combine source sense embeddings that cover different sets of word senses. Experimental results on Word Sense Disambiguation (WSD) and Word-in-Context (WiC) tasks show that the proposed meta-sense embedding method consistently outperforms several competitive baselines.

1 Introduction

In contrast to static word embedding methods (Mikolov et al., 2013a; Pennington et al., 2014) that learn a vector, that represents the meaning of a word, sense embedding methods (Loureiro and Jorge, 2019a; Camacho-Collados and Pilehvar, 2018; Scarlini et al., 2020a,b) learn multiple vectors per word, corresponding to the different senses of an ambiguous word. Prior work has shown that sense embeddings are useful for tasks such as Word Sense Disambiguation (WSD) and sense discrimination tasks such as Word in Context (WiC) (Loureiro and Jorge, 2019b; Pilehvar and Camacho-Collados, 2019). However, existing sense embeddings are trained on diverse resources such as sense tagged corpora or dictionary glosses, with varying levels of sense coverage (e.g. fully-covering all synsets in the WordNet or only a subset), and using different methods. Therefore, the performance reported by the existing sense embeddings on different downstream tasks and datasets vary significantly for different part-of-speech (PoS) categories. Moreover, it is not readily clear which sense embedding learning method should be used for disambiguating words in a given domain.

Meta-embedding learning has been successfully used to learn accurate and high coverage word- and sentence-level meta-embeddings by combining independently trained multiple source embeddings (Bollegala and O'Neill, 2022; Yin and Schütze, 2016a). However, to the best of our knowledge, meta-embedding learning methods have *not* been applied for sense embeddings before. Compared to word-level meta-embedding, sense-level meta-embedding has two important challenges.

Challenge 1 (*missing senses*). Compared to learning meta-word embeddings, where each word is assigned a single embedding, in static sense embeddings an ambiguous word is associated with multiple sense embeddings, each corresponding to a distinct sense of the ambiguous word. However, not all of the different senses of a word might be equally covered by all source sense embeddings.

Challenge 2 (*Misalignment between sense and context embeddings*). In downstream tasks such as WSD, we must determine the correct sense s of an ambiguous word w in a given context (i.e. a sentence) c. This is done by comparing the sense embeddings for each distinct sense of w against the context embedding of c, for example, computed using a Masked Language Model (MLM) such as BERT (Devlin et al., 2019). The sense corresponding to the sense embedding that has the maximum similarity with the context embedding is then selected as the correct sense of w in c. For

Findings of the Association for Computational Linguistics: ACL 2023, pages 2638–2651

sense embeddings such as LMMS (Loureiro and Jorge, 2019a) or ARES (Scarlini et al., 2020b) this is trivially achieved because they are both BERT-based embeddings and the cosine similarity between those sense embeddings and BERT embeddings can be directly computed. However, this is *not* the case for the meta-sense embeddings that exist in a different vector space than the context embeddings produced by BERT, where a projection between meta-sense and context embedding spaces must be learned before conducting WSD.

To address these challenges, we propose **Neighbourhood Preserving Meta-Sense Embedding** (NPMS) by incorporating multiple independently trained **source** sense embeddings to learn a **meta**-sense embedding such that the sense-related information captured by the source (input) sense embeddings is preserved in the (output) meta-sense embedding. NPMS can combine full-coverage sense embeddings with partial-coverage ones, thereby improving the sense coverage in the latter.

NPMS does *not* compare the source embeddings directly but require the nearest neighbours computed using source and meta sense embeddings to be similar. We call this *information preservation* criteria, and use Pairwise Inner Product (PIP) to compare the similarity distributions (nearest neighbours) over senses between meta and source embedding spaces. This allows us to address Challenge 1 using shared neighbours to compute the alignment between source and meta embedding spaces, without predicting any missing sense embeddings. To address Challenge 2, NPMS requires meta-sense embedding of a word sense to be similar to the contextualised (word) embeddings of the words that co-occur in the same sentence. We call this *contextual alignment*, and learn the sense-specific projection matrices that satisfy this criteria. This ensures that meta-sense embeddings could be used in downstream tasks such as WSD or WiC, where we must select the correct sense of an ambiguous word given its context.

We evaluate NPMS on WiC and WSD tasks against several competitive baselines for meta-embedding. Experimental results show that NPMS consistently outperforms all other methods in both tasks. More importantly, we obtain state-of-the-art (SoTA) performance for WSD and WiC tasks, reported by any static sense embedding method. Source code for the proposed method is publicly available.[1]

2 Related Work

Our work is related to both static sense embeddings and meta-embedding learning as we review next.

Static Sense Embeddings assign multiple embeddings for a single word, corresponding to its distinct senses. 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 a fixed number of clusters to represent a word, whereas Neelakantan et al. (2014) proposed a non-parametric model, which estimates the number of senses dynamically per each word. Chen et al. (2014) initialised sense embeddings by means of glosses from WordNet, and adapted the skip-gram objective (Mikolov et al., 2013b) to learn and improve sense embeddings jointly with word embeddings. Rothe and Schütze (2015) used pretrained word2vec embeddings to compose sense embeddings from sets of synonymous words. Camacho-Collados et al. (2016) created sense embeddings using structural knowledge from the BabelNet (Navigli and Ponzetto, 2010). Loureiro and Jorge (2019a) constructed sense embeddings by taking the average over the contextualised embeddings of the sense annotated tokens from SemCor. Scarlini et al. (2020a) used the lexical-semantic information in BabelNet to produce sense embeddings without relying on sense-annotated data. Scarlini et al. (2020b) also proposed ARES, a knowledge-based approach for constructing BERT-based embeddings of senses by means of the lexical-semantic information in BabelNet and Wikipedia.

Meta embedding learning was first proposed for combining multiple pretrained static word embeddings (Yin and Schütze, 2016b). Vector concatenation (Bollegala, 2022) is known to be a surprisingly strong baseline but increases the dimensionality of the meta-embedding with more sources. Coates and Bollegala (2018) showed that averaging performs comparable to concatenation under certain orthonormal conditions, while not increasing the dimensionality. Learning orthogonal projections prior to averaging has shown to further improve performance (Jawanpuria et al., 2020). Globally

[1] https://github.com/LivNLP/NPMS

linear (Yin and Schütze, 2016b), locally linear (Bollegala and Bao, 2018a) and autoencoder-based non-linear projections (Bollegala and Bao, 2018b) have been used to learn word-level meta-embeddings. Meta-embedding methods have been used for contextualised word embeddings (Kiela et al., 2018) and sentence embeddings (Takahashi and Bollegala, 2022; Poerner et al., 2020). For an extensive survey on meta-embedding learning see Bollegala and O'Neill (2022). However, to our best knowledge, we are the first to apply meta-embedding learning methods to learn sense embeddings.

3 Meta-Sense Embedding Learning

To explain our proposed method in detail, let us first consider a vocabulary $\mathcal{V}$ of words $w \in \mathcal{V}$. We further assume that each word w is typically associated with one or more distinct senses s and the set of senses associated with w is denoted by $\mathcal{S}_w$. In meta-sense embedding learning, we assume a sense s of a word to be represented by a set of n source sense embeddings. Let us denote the j-th source embedding of s by $\boldsymbol{x}_j(s) \in \mathbb{R}^{d_j}$, where d_j is the dimensionality of the j-th source embedding.

We project the j-th source embedding by a matrix $\mathbf{P}_j \in \mathbb{R}^{d \times d_j}$ into a common meta-sense embedding space with dimensionality d. The meta-sense embedding, $\boldsymbol{m}(s) \in \mathbb{R}^d$ of s is computed as the unweighted average of the projected source sense embeddings as given by (1).

$$\boldsymbol{m}(s) = \frac{1}{n} \sum_{j=1}^{n} \mathbf{P}_j \boldsymbol{x}_j(s) \tag{1}$$

After this projection step, all source sense embeddings live in the same d-dimensional vector space, thus enabling us to add them as done in (1).

An advantage of considering the average of the projected source embeddings as the meta-sense embedding is that, even if a particular sense is not covered by one or more source sense embeddings, we can still compute a meta-sense embedding using the remainder of the source sense embeddings. Moreover, prior work on word-level and sentence-level meta-embedding have shown that averaging after a linear projection improves performance when learning meta embeddings (Coates and Bollegala, 2018; Jawanpuria et al., 2020; Poerner et al., 2020).

If we limit the projection matrices to be orthonormal, they can be seen as optimally rotating the source sense embeddings such that the projected source embeddings could be averaged in the meta-embedding space. However, we observed that dropping this regularisation term to produce better meta-sense embeddings in our experiments. Therefore, we did not impose any orthonormality restrictions on the projection matrices.

We require a meta-sense embedding to satisfy two criteria: (a) **sense information preservation** and (b) **contextual alignment**. The two criteria jointly ensure that the meta-sense embeddings we learn are accurate and can be used in downstream tasks such as WSD in conjunction with contextualised word embeddings produced by an MLM. Next, we describe each of those criteria in detail.

3.1 Sense Information Preservation

Given that the individual source sense embeddings are trained on diverse sense-related information sources, we would like to preserve this information as much as possible in the meta-sense embeddings we create from those source sense embeddings. This is particularly important in meta-embedding learning because we might not have access to all the resources that were used to train the individual source sense embeddings, nor we will be training meta-embeddings from scratch but will be relying upon pretrained sense embeddings as the sole source of sense-related information into the meta-embedding learning process. Therefore, we must preserve the complementary sense-related information encoded in the source sense embeddings as much as possible in their meta-sense embedding.

It is not possible however to directly compare the meta-sense embeddings computed using (1) against the source sense embeddings because they have different dimensionalities and live in different vector spaces. This makes it challenging when quantifying the amount of information lost due to meta embedding using popular loss functions such as squared Euclidean distance between source and meta embeddings. To address this problem we resort to PIP, which has been previously used to determine the optimal dimensionality of word embeddings (Yin and Shen, 2018) and learning concatenated word-level meta embeddings (Bollegala, 2022).

Given a source/meta embedding matrix $\mathbf{E}$, the corresponding PIP matrix is given by (2)

$$\mathrm{PIP}(\mathbf{E}) = \mathbf{E}\mathbf{E}^\top \tag{2}$$

Specifically, PIP matrix contains the inner-products between all pairs of sense embeddings

represented by the rows of $\mathbf{E}$. $\mathrm{PIP}(\mathbf{E})$ is a symmetric matrix with its number of rows (columns) equal to the total number of unique senses covering all the words in the vocabulary. PIP matrices can be efficiently computed for larger dimensions and vocabularies because the inner-product computation can be parallelized over the embeddings.

Let us denote the source sense embedding matrix for the j-th source by $\mathbf{X}_j$, where the i-th row represents sense embedding $\boldsymbol{x}_j(s_i)$ learnt for the i-th sense s_i. Likewise, let us denote by $\mathbf{M}$ the meta-sense embedding matrix, where the i-th row represents the meta-sense embedding $\boldsymbol{m}(s_i)$ computed for s_i using (1). Because the shape of PIP matrices are independent from the dimensionalities of the embedding spaces, and the rows are aligned (i.e. sorted by the sense ids s_i), we can compare the meta-sense embedding against the individual source sense embedding using PIP loss, L_{pip}, given by (3).

$$L_{\text{pip}} = \sum_{j=1}^{n} ||\mathrm{PIP}(\mathbf{X}_j) - \mathrm{PIP}(\mathbf{M})||_F^2 \quad (3)$$

Here, $||\mathbf{A}||_F = \sqrt{\sum_{l,m} a_{lm}^2}$ denotes the Frobenius norm of the matrix $\mathbf{A}$. PIP loss can be seen as comparing the distributions of similarity scores computed using the meta-sense embedding and each of the individual source sense embeddings for the same set of senses. Although the actual vector spaces might be different and initially not well-aligned due to the projection and averaging steps in (1), we would require the neighbourhoods computed for each word to be approximately similar in the meta-sense embedding space and each of the source sense embedding spaces. PIP loss given in (3) measures this level of agreement between meta and source embedding spaces.

3.2 Contextual Alignment

The context in which an ambiguous word has been used provides useful clues to determine the correct sense of that word (Zhou and Bollegala, 2021). For example, consider the following two sentences: **(S1)** *I went to the bank to withdraw some cash.*, and **(S2)** *The river bank was crowded with people doing BBQs.* Words *cash* and *withdraw* indicate the *financial institute* sense of bank in **S1**, whereas the words *river*, *BBQ* indicate the *sloping land* sense of bank in **S2**.

Let us denote the contextualised word embedding of a word w in a context c by $\boldsymbol{f}(w;c)$. MLMs such as BERT and RoBERTa (Liu et al., 2019) have been used in prior work in WSD to compute context-sensitive representations for ambiguous words. Then, the above-described agreement between the sense s of w and its context c can be measured by the similarity between the meta-sense embedding $\boldsymbol{m}(s)$ and the contextualised embedding $\boldsymbol{f}(w;c)$. We refer to this requirement as the *contextual alignment* between a meta-sense embedding and contextualised word embeddings.

Given a sense annotated dataset such as SemCor, we represent it by a set $\mathcal{T}$ of tuples (w,s,c), where the word w is annotated with its correct sense s in context c. Then, we define the contextual alignment loss L_{cont} as (negative) average cosine similarity between $\boldsymbol{m}(s)$ and $\boldsymbol{f}(w;c)$, given by (4).

$$L_{\text{cont}} = -\sum_{(w,s,c)\in\mathcal{T}} \frac{\boldsymbol{m}(s)^\top \boldsymbol{f}(w;c)}{||\boldsymbol{m}(s)||_2 \, ||\boldsymbol{f}(w;c)||_2} \quad (4)$$

Minimising the contextual alignment loss in (4), will maximise the cosine similarity between the meta-sense embedding and the corresponding contextualised embedding.

In contrast to the PIP-loss defined by (3), which can be computed without requiring sense annotated data, the contextual alignment loss defined by (4) requires sense annotated data. However, SemCor, the sense annotated dataset that we use for computing the contextual alignment loss in this paper, is already being used by many existing pretrained source sense embeddings. Therefore, we emphasise that we are *not* requesting for any additional training resources during the meta-sense embedding learning process beyond what has been already used to train the source sense embeddings. Moreover, ablation studies (§5) show that PIP-loss alone obtains significant improvements, without the contextual alignment loss.

Contextual alignment loss can also be motivated from an application perspective. Sense embeddings are often used to represent word senses in downstream tasks such as WSD. A typical approach for predicting the sense of an ambiguous word w as used in a given context c is to measure the cosine similarity between each sense embedding of w and the context embedding for c (Scarlini et al., 2020b; Loureiro and Jorge, 2019a). The objective given in (4) can be seen as enforcing this property directly into the meta-sense embedding learning process. As we later see in §4, NPMS perform particularly well in WSD benchmarks.

In order to be able to compute the cosine similarity between meta-sense embeddings and contextualised word embeddings, we must first ensure that they have the same dimensionality. This can be achieved by either (a) setting the dimensionality of the meta-sense embeddings equal to that of the contextualised word embeddings, or (b) by learning a projection matrix that adjusts the dimensionality of the meta-sense embeddings to that of the contextualised word embeddings.

3.3 Parameter Learning

We consider the linearly-weighted sum of the PIP-loss and contextual alignment loss as the total loss, L_{tot}, given by (5).

$$L_{\text{tot}}(\{\mathbf{P}_j\}_{j=1}^{n}) = \alpha L_{\text{pip}} + (1-\alpha)L_{\text{cont}} \quad (5)$$

Here, the parameters to be learnt are the projection matrices $\mathbf{P}_j$ for the sources $j = 1, \ldots, n$. The weighting coefficient $\alpha \in [0, 1]$ is a hyperparameter determining the emphasis between the two losses. In our experiments, we tune α using validation set of the Senseval-3 WSD dataset (Snyder and Palmer, 2004).

Compared to the cosine similarity, which is upper bounded by 1, the PIP-loss grows with the size of the PIP matrices being used. Therefore, we found that scaling the two losses by their mean values to be important to stabilise the training. We initialise the projection matrices to the identity matrix and use vanilla stochastic gradient descent with a learning rate of 0.001, determined using the validation set of the Senseval-3 WSD dataset.

4 Experiments

4.1 Source Embeddings

Our proposed NPMS is agnostic to the methods used to learn the source sense embeddings, and thus in principle can be used to meta-embed any source sense embedding. In our experiments, we use the following source sense embeddings because of their accuracy, public availability, coverage word senses and diversity (i.e. trained on different resources to have different dimensionalities) such that we can conduct an extensive evaluation.

LMMS (Loureiro and Jorge, 2019a) (Language Modelling Makes Sense) is a supervised approach to learn full-coverage static sense embeddings that cover all of the 206,949 senses in the WordNet. We use three variants of LMMS (Loureiro et al., 2022) embeddings[2] as sources in our experiments: (a) **LMMS** (uses 1024 dimensional `bert-large-cased`[3] embeddings with semantic networks (i.e., WordNet) and glosses to create 2048 dimensional sense emebddings), (b) **LMMS (XLNet)** (uses 1024 dimensional `xlnet-large-cased`[4] as the base MLM, and averages contextualised embeddings computed from SemCor and WordNet glosses), and (c) **LMMS (RoBERTa)** (uses 1024 dimensional `roberta-large`[5] as the base MLM, and and averages contextualised embeddings computed from SemCor and WordNet glosses).

SenseEmBERT (Scarlini et al., 2020a) (Sense Embedded BERT) obviates the need for sense-annotated corpora by using the BabelNet[6] mappings between WordNet senses and Wikipedia pages to construct sense embeddings with 2048 dimensions, covering all the 146,312 English nominal senses in the WordNet. Each sense embedding consists of two components: (a) the average of the word embedding of the a target sense's relevant words, and (b) the average of the BERT encoded tokens of the sense gloss. For the brevity of the notation, we denote SenseEmBERT as **SBERT** in the remainder of this paper.

ARES (Scarlini et al., 2020b) (context-AwaRe EmbeddingS) is a semi-supervised method that learns sense embeddings with full-coverage of the WordNet and is 2048 dimensional. ARES embeddings are created by applying BERT on the glossary information and the information contained in the SyntagNet (Maru et al., 2019). It outperforms LMMS in WSD benchmarks.

DeConf (Pilehvar and Collier, 2016) are the 50-dimensional[7] De-conflated Semantic Embeddings created from Wikipedia and Gigaword corpus using GloVe (Pennington et al., 2014). DeConf enables us to evaluate the effect of combining a source that has significantly smaller dimensionality than the other source sense embeddings.

The intersection of the LMMS_{2048} and ARES contains 206,949 senses, which is equivalent to the total number of senses in the WordNet because they both cover all the sense in the WordNet (i.e.

[2] https://github.com/danlou/LMMS
[3] https://huggingface.co/bert-large-cased
[4] https://huggingface.co/xlnet-large-cased
[5] https://huggingface.co/roberta-large
[6] babelnet.org
[7] https://pilehvar.github.io/deconf/

full coverage sense embeddings). On the other hand, the intersection between the LMMS_{2048} and SensEmBERT as well as the intersection between the ARES and SensEmBERT contains 146,312 senses, which is the total number of nominal senses in the WordNet. By using source sense embeddings with different sense coverages we aim to evaluate the ability of meta-sense embedding methods to learn accurate sense embeddings by exploiting the complementary strengths in the sources.

4.2 Evaluation Tasks

We compare the accuracy of meta-sense embeddings using two standard tasks that have been used in prior work on sense embedding learning.

Word Sense Disambiguation (WSD): WSD is a longstanding problem in NLP, which aims to assign an ambiguous word in a context with a word sense (Navigli, 2009). To test whether NPMS can disambiguate the different senses of an ambiguous word, we conduct a WSD task using the evaluation framework proposed by Raganato et al. (2017), which contains all-words English WSD datasets: Senseval-2 (**SE2**; Edmonds and Cotton, 2001), Senseval-3 (**SE3**; Snyder and Palmer, 2004), SemEval-07 (**SE07**; Pradhan et al., 2007), SemEval-13 (**SE13**; Navigli et al., 2013) and SemEval-15 (**SE15**; Moro and Navigli, 2015). We use the official framework to avoid any discrepancies in the scoring methodology.

We perform WSD following the 1-NN procedure, where we compute the contextualised embedding, $\boldsymbol{f}(w; c)$, produced using an MLM.[8] We then measure the cosine similarity, $\phi(\boldsymbol{m}(s), \boldsymbol{f}(w; c))$, between the source/meta sense embedding for each sense s of w, $\boldsymbol{m}(s)$, and $\boldsymbol{f}(w; c)$, and select the sense with the maximum cosine similarity as the correct sense of w in c.

Word-in-Context (WiC): WiC is framed as a binary classification task, where given a target word w and two contexts c_2 and c_2, the objective is to determine if w occurring in c_1 and c_2 carries the same meaning. A method that assigns the same vector to all senses of w would report a chance-level (i.e. 50%) accuracy on WiC.

Given a target word w in two contexts c_1 and c_2, we first determine the meta-sense embeddings of w, which are $\boldsymbol{m}(s_1)$ and $\boldsymbol{m}(s_2)$ corresponding to the senses of w used in respectively c_1 and c_2. Let the contextualised word embedding of w in c_1 and c_2 respectively be $\boldsymbol{f}(w; c_1)$ and $\boldsymbol{f}(w; c_2)$. We train a binary logistic regression classifier on the WiC training set. Following the work from Zhou and Bollegala (2021), we use the cosine similarities between the two vectors in the following six pairs as features: $\phi(\boldsymbol{m}(s_1), \boldsymbol{m}(s_2))$, $\phi(\boldsymbol{f}(w; c_1), \boldsymbol{f}(w; c_2))$, $\phi(\boldsymbol{m}(s_1), \boldsymbol{f}(w; c_1))$, $\phi(\boldsymbol{m}(s_2), \boldsymbol{f}(w; c_2))$, $\phi(\boldsymbol{m}(s_1), \boldsymbol{f}(w; c_2))$ and $\phi(\boldsymbol{m}(s_2), \boldsymbol{f}(w; c_1))$.

4.3 Meta-Embedding Methods

We extend prior works on word-level meta-embedding learning to meta-sense embedding learning by taking the sense embeddings described in §4.1 as source embeddings, and compare them with **NPMS** embeddings. We compare against the following methods:

- **AVG** (Coates and Bollegala, 2018) takes the average over the embeddings of a sense from different sources embeddings.
- **CONC** (Yin and Schütze, 2016a) creates meta-embeddings by concatenating the embeddings from different source embeddings.
- **SVD** (Yin and Schütze, 2016a) performs dimensionality reduction on the concatenated source embeddings.
- **AEME** (Bollegala and Bao, 2018a) is an autoencoder-based method for meta-embedding learning, which is the current SoTA unsupervised word-level meta-embedding learning method.

We use 2048 output dimensions for both SVD and AEME in the experiments, determined to be the best for those methods on validation data.

As noted in § 4.2, both WSD and WiC tasks require us to compute the cosine similarity, ϕ, between a source/meta sense embedding, $\boldsymbol{m}(s)$, of a sense s and a contextualised word embedding, $\boldsymbol{f}(w; c)$, of the ambiguous word w in context c. However, unlike for NPMS, which explicitly guarantees that its meta-sense embeddings are directly comparable with the contextualised word embeddings via the contextual loss (4), in general, the meta-sense embeddings produced by other methods do not always exist in the contextualised word embedding space associated with the MLM, which requires careful consideration as discussed next.

[8] In the case of BERT, we average the last four layers for each word w in a test sentence c.

As an concrete example, let us consider the meta-embedding of the three sources LMMS, ARES and SenseEmBERT, all of which are 2048 dimensional and computed by concatenating two 1024-dimensional BERT embeddings, averaged over different lexical resources. Therefore, using the same 1024-dimensional BERT embeddings and by concatenating $\boldsymbol{f}(w;c)$ twice, we can obtain a 2048-dimensional BERT-based contextualised embedding for w that can be used to compute the cosine similarity with a source sense embedding in this case. We consider the meta-embedding of source sense embeddings with different dimensionalities and MLMs other than BERT such as LMMS (XLNet), LMMS (RoBERTa) and DeConf later in our experiments.

Next, let us consider the meta-sense embeddings produced by CONC. Because the inner-product decomposes trivially over vector concatenation, we can copy and concatenate $\boldsymbol{f}(w;c)$ to match $\boldsymbol{m}(s)$ produced by CONC. For example, if CONC is used with LMMS and ARES, we can concatenate $\boldsymbol{f}(w;c)$ four times, and then compute the inner-product with the meta-sense embedding. AVG does not change the dimensionality of the meta-sense embedding space. Therefore, we only need to concatenate $\boldsymbol{f}(w;c)$ twice when computing the cosine similarity with AVG for any number of source sense embeddings.

Unfortunately, the meta-sense embedding spaces produced by SVD and AEME are not directly comparable against that of contextualised embeddings due to the differences in dimensionality and non-linear transformations introduced (cf. AEME uses autoencoders). Therefore, we learn a projection matrix, $\mathbf{A}$, between $\boldsymbol{m}(s)$ and $\boldsymbol{f}(w;c)$ by minimising the squared Euclidan distance given by (6), computed using the SemCor training dataset, $\mathcal{T}$.

$$\sum_{(w,s,c)\in\mathcal{T}} ||\mathbf{A}\boldsymbol{m}(s) - \boldsymbol{f}(w;c)||_2^2 \qquad (6)$$

After training, we compute the cosine similarity, $\phi(\mathbf{A}\boldsymbol{m}(s), \boldsymbol{f}(w;c))$, between the transformed SVD and AEME meta-sense embedding and contextualised embeddings.

5 Results

5.1 Effect of Meta-embedding Learning

Table 1 compares the performance of NPMS against the meta-embedding methods described in §4.3 on WSD and WiC. We see that NPMS obtains the overall best performance for WSD (ALL) as well as on WiC. Among the three sources, ARES reports the best performance for WSD (ALL), while SBERT does so for WiC. In SE2, SE07 datasets NPMS reports the best performance, whereas AVG, SBERT and ARES do so respectively in SE3, SE13 and SE15. Among the baseline methods, we see AVG to report the best results, which is closely followed by CONC. Poor performance of SVD shows the challenge of applying dimensionality reduction methods on CONC due to missing sense embeddings. Although AEME has reported the SoTA performance for word-level meta-embedding, applying it directly on sense embeddings is suboptimal. This shows the difference between word- vs. sense-level meta-embedding learning problems, and calls for sense-specific meta-embedding learning methods.

According to the WiC leader board,[9] the performance reported by NPMS is second only to SenseBERT (Levine et al., 2020), which is a contextualised sense embedding method obtained by fine tuning BERT on WordNet supersenses. Therefore, the performance of NPMS can be seen as the SoTA for any *static* sense embedding method.

5.2 Effect of Source Embeddings

The performance of a meta-embedding depends on the source embeddings used. Therefore, we evaluate the ability of NPMS to create meta-sense embeddings from diverse source sense embeddings that have different dimensionalities and created from different MLMs. Due space limitations, in Table 2 we compare NPMS against AVG, which reported the best performance among all other meta-embedding learning methods in Table 1. From Table 2, we see that when the dimensionalities of the two source sense embeddings are identical (i.e. 2048 dimensional LMMS + ARES or LMMS + SBERT configurations) or similar (i.e. 2048 dimensional ARES + 2048 dimensional SBERT configuration), AVG closely matches the performance of NPMS in WSD and WiC evaluations. However, we see a drastically different trend when the two sources are not BERT-based (e.g. XLNet, RoBERTa) or when they have significantly different dimensionalities (1024 dimensional LMMS (XLNet), LMMS (RoBERTa) and 50 dimensional DeConf). In such settings, we see that NPMS to performs significantly better than AVG across all

[9] https://pilehvar.github.io/wic/

	SE2	SE3	SE07	SE13	SE15	ALL	WiC
LMMS	76.34	75.57	68.13	75.12	77.01	75.44	69.30
ARES	78.05	77.08	70.99	77.31	**83.17**	77.91	68.50
SBERT	53.11	52.22	41.37	**78.77**	55.12	59.85	71.14
AVG	79.36	**77.46**	70.33	77.86	80.82	78.17	71.16
CONC	78.22	77.14	70.99	77.37	82.97	77.97	70.38
SVD	75.02	74.22	67.25	72.81	74.85	73.80	63.01
AEME	78.53	76.92	69.01	76.09	78.96	77.03	70.69
NPMS	**79.93**	77.30	**71.65**	77.49	81.21	**78.37**	**71.47**

Table 1: F1 scores on WSD benchmarks and accuracy on WiC are shown for the three sources (top) and for the different meta-embedding methods (bottom).

	SE2	SE3	SE07	SE13	SE15	ALL	WiC
LMMS(BERT) [2048] + ARES (BERT) [2048]							
AVG	**78.79**	77.03	69.89	77.13	**81.80**	77.83	70.22
NPMS	78.53	**77.14**	**71.87**	**77.37**	81.60	**77.93**	70.22
ARES (BERT) [2048] + SBERT [2048]							
AVG	78.57	77.35	71.21	78.10	**81.70**	78.13	71.32
NPMS	**78.79**	77.41	**71.65**	**78.53**	81.41	**78.30**	71.32
LMMS (BERT) [2048] + SBERT [2048]							
AVG	77.70	76.16	68.79	78.04	77.69	76.82	69.59
NPMS	**78.05**	**76.86**	**69.89**	**78.28**	**78.28**	**77.32**	**71.79**
LMMS(XLNet) [1024] + DeConf [50]							
AVG	40.80	35.68	21.32	41.61	43.93	38.89	66.46
NPMS	**50.88**	**41.68**	**40.66**	**53.04**	**53.13**	**48.70**	**69.26**
LMMS(RoBERTa) [1024] + DeConf [50]							
AVG	39.35	34.97	26.15	41.48	42.47	38.33	66.46
NPMS	**48.77**	**44.81**	**39.34**	**53.41**	**53.52**	**48.89**	**69.75**

Table 2: Meta-sense embedding of sources with different dimensionalities (shown in brackets) and MLMs.

WSD benchmarks as well as on WiC. Recall that AVG assumes (a) the source embedding spaces to be orthogonal, and (b) applies zero-padding to the smaller dimensional source embeddings to make them aligned with the rest of the source embeddings. Both of those assumptions do not hold true when the source embeddings are created from diverse MLMs or have significantly different numbers of dimensions, which leads to suboptimal performances in AVG. On the other hand, NPMS does *not* directly compare source sense embeddings, but instead consider neighbourhoods computed from the source sense embeddings. Moreover, zero-padding is not required in NPMS because the contextual alignment step ensures the proper alignment between the contextual embedding and meta-sense embedding spaces. These advantages of NPMS are clearly evident from Table 2.

Method	WSD (ALL)	WiC
SVD with proj.	74.80	66.93
SVD without proj.	35.90	60.34
AEME with proj.	76.02	68.65
AEME without proj.	41.60	53.61

Table 3: Effect of learning a projection matrix between meta-sense vs. BERT embedding spaces.

5.3 Effect of Projection Learning

Table 3 shows the importance of learning a projection matrix via (6) between meta-sense and contex-

	SE2	SE3	SE07	SE13	SE15	ALL	WiC
Both	**79.93**	**77.30**	71.65	77.49	**81.21**	**78.37**	**71.47**
L_{pip} only	79.80	77.03	**71.87**	77.49	80.72	78.20	70.69
L_{cont} only	79.54	77.19	70.77	**77.86**	80.33	78.12	71.32

Table 4: Ablation between the PIP-loss (L_{pip}) and contextual alignment loss (L_{cont}).

tualised embeddings, for SVD and AEME. We see that the performance of both of those methods drop significantly without the projection matrix learning step. Even with projection matrices, SVD and AEME do not outperform simpler baselines such as AVG or CONC. On the other hand, NPMS does not require such a projection matrix learning step and consistently outperforms all those methods across multiple WSD and WiC benchmarks.

5.4 Effect of the Two losses

To understand the contributions of the two loss terms PIP-loss (L_{pip}) and contextual alignment loss (L_{cont}), we conduct an ablation study where we train NPMS with three sources using only one of the two losses at a time. From Table 4, we see that in both WiC and WSD (ALL, SE2, SE3, SE15), the best performance is obtained by using both losses. Each loss contributes differently in different datasets, although the overall difference between the two losses is non-significant (according to a paired Student's t-test with $p < 0.05$). This is particularly encouraging because PIP-loss can be computed without having access to a sense labelled corpus such as SemCor. Such resources might not be available in specialised domains such as medical or legal texts. Therefore, in such cases we can still apply NPMS trained using only the PIP-loss. Although we considered a linearly-weighted combination of the two losses in (5), we believe further improvements might be possible by exploring more complex (nonlinear) combinations of the two losses. However, exploring such combinations is beyond the scope of current paper and is deferred to future work.

6 Conclusion

We proposed the first-ever meta-sense embedding learning method. Experimental results on WiC and WSD datasets show that our proposed NPMS surpasses previously published results for static sense embedding, and outperforms multiple word-level meta-embedding learning methods when applied to sense embeddings. Our evaluations were limited to English and we will consider non-English sense embeddings in our future work.

7 Limitations

All source sense embeddings we used in our experiments are only covering the English language, which is morphologically limited. Therefore, it is unclear whether our results and conclusions will still be valid for meta-sense embeddings created for languages other than English. On the other hand, there are WSD and WiC benchmarks for other languages such as SemEval-13, SemEval-15, XL-WSD (Pasini et al., 2021) and WiC-XL (Raganato et al., 2020), as well as multilingual sense embeddings such as ARES$_m$ (Scarlini et al., 2020b) and SensEmBERT (Scarlini et al., 2020a). Extending our evaluations to cover multilingual sense embeddings is deferred to future work.

Our meta-sense embedding method requires static sense embeddings, and cannot be applied to contextualised sense embedding methods such as SenseBERT (Levine et al., 2020). There have been some work on learning word-level and sentence-level (Takahashi and Bollegala, 2022; Poerner et al., 2020) meta-embeddings using contextualised word embeddings produced by MLMs as the source embeddings. However, contextualised sense embedding methods are limited compared to the numerous static sense embedding methods. This is partly due to the lack of large-scale sense annotated corpora, required to train or fine-tune contextualised sense embeddings. Extending our work to learn meta-sense embeddings using contextualised word embeddings as source embeddings is an interesting future research direction.

8 Ethical Considerations

We compared our proposed method, NPMS, with several baselines on WSD and WiC tasks. In this work, we did not annotate any datasets by ourselves and used corpora and benchmark datasets that have been collected, annotated and repeatedly used for

evaluations in prior works. To the best of our knowledge, no ethical issues have been reported concerning these datasets. Nevertheless, prior work from Zhou et al. (2022) shows that pretrained sense embeddings encode various types of social biases such as gender and racial biases. Moreover, it has also been reported recently that word-level meta-embedding methods can amplify the social biases encoded in the source embeddings (Kaneko et al., 2022). Therefore, we emphasise that it is important to evaluate the meta-sense embeddings learnt in this work for unfair social biases before they are deployed to downstream applications.

Acknowledgements

Danushka Bollegala holds concurrent appointments as a Professor at University of Liverpool and as an Amazon Scholar. This paper describes work performed at the University of Liverpool and is not associated with Amazon.

References

Danushka Bollegala. 2022. Learning meta word embeddings by unsupervised weighted concatenation of source embeddings. In *Proc. of the 31st International Joint Conference on Artificial Intelligence (IJCAI-ECAI)*.

Danushka Bollegala and Cong Bao. 2018a. Learning word meta-embeddings by autoencoding. In *Proceedings of the 27th international conference on computational linguistics*, pages 1650–1661.

Danushka Bollegala and Cong Bao. 2018b. Learning word meta-embeddings by autoencoding. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1650–1661, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Danushka Bollegala and James O'Neill. 2022. A survey on word meta-embedding learning. In *Proc. of the 31st International Joint Conference on Artificial Intelligence (IJCAI-ECAI)*.

Jose Camacho-Collados and Mohammad Taher Pilehvar. 2018. From word to sense embeddings: A survey on vector representations of meaning. *J. Artif. Int. Res.*, 63(1):743–788.

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

Xinxiong Chen, Zhiyuan Liu, and Maosong Sun. 2014. A unified model for word sense representation and disambiguation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1025–1035, Doha, Qatar. Association for Computational Linguistics.

Joshua Coates and Danushka Bollegala. 2018. Frustratingly easy meta-embedding – computing meta-embeddings by averaging source word embeddings. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 194–198, New Orleans, Louisiana. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Philip Edmonds and Scott Cotton. 2001. SENSEVAL-2: Overview. In *Proceedings of SENSEVAL-2 Second International Workshop on Evaluating Word Sense Disambiguation Systems*, pages 1–5, Toulouse, France. Association for Computational Linguistics.

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.

Pratik Jawanpuria, Satya Dev N T V, Anoop Kunchukuttan, and Bamdev Mishra. 2020. Learning geometric word meta-embeddings. In *Reps4NLP*, pages 39–44, Online.

Masahiro Kaneko, Danushka Bollegala, and Naoaki Okazaki. 2022. Gender bias in meta-embeddings. In *Proc. of 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP 2022)*.

Douwe Kiela, Changhan Wang, and Kyunghyun Cho. 2018. Dynamic meta-embeddings for improved sentence representations. In *EMNLP*, pages 1466–1477.

Yoav Levine, Barak Lenz, Or Dagan, Ori Ram, Dan Padnos, Or Sharir, Shai Shalev-Shwartz, Amnon Shashua, and Yoav Shoham. 2020. SenseBERT: Driving some sense into BERT. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4656–4667, Online. Association for Computational Linguistics.

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.

Daniel Loureiro and Alípio Jorge. 2019a. Language modelling makes sense: Propagating representations through WordNet for full-coverage word sense disambiguation. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*, pages 5682–5691, Florence, Italy. Association for Computational Linguistics.

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

Daniel Loureiro, Alípio Mário Jorge, and Jose Camacho-Collados. 2022. LMMS reloaded: Transformer-based sense embeddings for disambiguation and beyond. *Artificial Intelligence*, 305:103661.

Marco Maru, Federico Scozzafava, Federico Martelli, and Roberto Navigli. 2019. SyntagNet: Challenging supervised word sense disambiguation with lexical-semantic combinations. 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 3534–3540, Hong Kong, China. Association for Computational Linguistics.

Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, pages 3111–3119, Red Hook, NY, USA. Curran Associates Inc.

Andrea Moro and Roberto Navigli. 2015. SemEval-2015 task 13: Multilingual all-words sense disambiguation and entity linking. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 288–297, Denver, Colorado. Association for Computational Linguistics.

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

Roberto Navigli, David Jurgens, and Daniele Vannella. 2013. SemEval-2013 task 12: Multilingual word sense disambiguation. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 222–231, Atlanta, Georgia, USA. Association for Computational Linguistics.

Roberto Navigli and Simone Paolo Ponzetto. 2010. Babelnet: Building a very large multilingual semantic network. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 216–225.

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.

Tommaso Pasini, Alessandro Raganato, and Roberto Navigli. 2021. Xl-wsd: An extra-large and cross-lingual evaluation framework for word sense disambiguation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 13648–13656.

Jeffrey Pennington, Richard Socher, and Christopher D 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.

Mohammad Taher Pilehvar and Jose Camacho-Collados. 2019. WiC: the word-in-context dataset for evaluating context-sensitive meaning representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1267–1273, Minneapolis, Minnesota. Association for Computational Linguistics.

Mohammad Taher Pilehvar and Nigel Collier. 2016. Deconflated semantic representations. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1680–1690, Austin, Texas. Association for Computational Linguistics.

Nina Poerner, Ulli Waltinger, and Hinrich Schütze. 2020. Sentence meta-embeddings for unsupervised semantic textual similarity. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7027–7034, Online. Association for Computational Linguistics.

Sameer Pradhan, Edward Loper, Dmitriy Dligach, and Martha Palmer. 2007. SemEval-2007 task-17: English lexical sample, SRL and all words. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 87–92, Prague, Czech Republic. Association for Computational Linguistics.

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

Alessandro Raganato, Tommaso Pasini, Jose Camacho-Collados, and Mohammad Taher Pilehvar. 2020. XL-WiC: A multilingual benchmark for evaluating semantic contextualization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7193–7206, Online. Association for Computational Linguistics.

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

Sascha Rothe and Hinrich Schütze. 2015. AutoExtend: Extending word embeddings to embeddings for synsets and lexemes. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1793–1803, Beijing, China. Association for Computational Linguistics.

Bianca Scarlini, Tommaso Pasini, and Roberto Navigli. 2020a. SensEmBERT: Context-Enhanced Sense Embeddings for Multilingual Word Sense Disambiguation. In *Proceedings of the Thirty-Fourth Conference on Artificial Intelligence*, pages 8758–8765. Association for the Advancement of Artificial Intelligence.

Bianca Scarlini, Tommaso Pasini, and Roberto Navigli. 2020b. With More Contexts Comes Better Performance: Contextualized Sense Embeddings for All-Round Word Sense Disambiguation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Benjamin Snyder and Martha Palmer. 2004. The English all-words task. In *Proceedings of SENSEVAL-3, the Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 41–43, Barcelona, Spain. Association for Computational Linguistics.

Keigo Takahashi and Danushka Bollegala. 2022. Unsupervised attention-based sentence-level meta-embeddings from contextualised language models. In *Proc. of LREC*.

Wenpeng Yin and Hinrich Schütze. 2016a. Learning word meta-embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1351–1360.

Wenpeng Yin and Hinrich Schütze. 2016b. Learning word meta-embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1351–1360, Berlin, Germany. Association for Computational Linguistics.

Zi Yin and Yuanyuan Shen. 2018. On the dimensionality of word embedding. In *Proc. of NeurIPS*, pages 887–898.

Yi Zhou and Danushka Bollegala. 2021. Learning sense-specific static embeddings using contextualised word embeddings as a proxy. In *Proceedings of the 35th Pacific Asia Conference on Language, Information and Computation*, pages 493–502, Shanghai, China. Association for Computational Lingustics.

Yi Zhou, Masahiro Kaneko, and Danushka Bollegala. 2022. Sense embeddings are also biased – evaluating social biases in static and contextualised sense embeddings. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1924–1935, Dublin, Ireland. Association for Computational Linguistics.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
section 6

☑ A2. Did you discuss any potential risks of your work?
section 7

☑ A3. Do the abstract and introduction summarize the paper's main claims?
abstract and section 1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

section 4

☑ B1. Did you cite the creators of artifacts you used?
section 3 and 4

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Not applicable. Left blank.

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
section 3 and 4

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
section 4

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
section 4

C ☑ Did you run computational experiments?

section 4

☐ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Not applicable. Left blank.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
section 3 and 4

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
section 4

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
section 4

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Not applicable. Left blank.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Not applicable. Left blank.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Not applicable. Left blank.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Not applicable. Left blank.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Not applicable. Left blank.