Distilling Word Meaning in Context from Pre-trained Language Models

Yuki Arase[1⋆] and **Tomoyuki Kajiwara**[2]
[1]Graduate School of Information Science & Technology, Osaka University, Japan
⋆Artificial Intelligence Research Center (AIRC), AIST, Japan
[2]Graduate School of Science and Engineering, Ehime University, Japan
arase@ist.osaka-u.ac.jp, kajiwara@cs.ehime-u.ac.jp

Abstract

In this study, we propose a self-supervised learning method that distils representations of word meaning in context from a pre-trained masked language model. Word representations are the basis for context-aware lexical semantics and unsupervised semantic textual similarity (STS) estimation. A previous study transforms contextualised representations employing static word embeddings to weaken excessive effects of contextual information. In contrast, the proposed method derives representations of word meaning in context while preserving useful context information intact. Specifically, our method learns to combine outputs of different hidden layers using self-attention through self-supervised learning with an automatically generated training corpus. To evaluate the performance of the proposed approach, we performed comparative experiments using a range of benchmark tasks. The results confirm that our representations exhibited a competitive performance compared to that of the state-of-the-art method transforming contextualised representations for the context-aware lexical semantic tasks and outperformed it for STS estimation.

1 Introduction

Word representations are the basis for various natural language processing tasks. Particularly, they are crucial as a component in context-aware lexical semantics and in the estimation of unsupervised semantic textual similarity (STS) (Arora et al., 2017; Ethayarajh, 2018; Yokoi et al., 2020). Word representations are desired to represent word meaning in context to improve these downstream tasks. Large-scale masked language models pre-trained on massive corpora, *e.g.*, bi-directional encoder representations from transformers (BERT) (Devlin et al., 2019), embed both the context and meaning of a word; thus, word-level representations generated by such masked language models are called contextualised word representations. Previous studies (Ethayarajh, 2019; Vulić et al., 2020) have revealed that lexical information and context-specific information are captured in different layers of masked language models. They argued that a sophisticated mechanism is required to derive representations of word meaning in context from them. Although contextualised word representations have shown considerable promise, how best to compose the outputs of different layers of masked language models to effectively represent word meaning in context remains an open question.

Liu et al. (2020) improved contextualised word representations by transforming their space towards static word embeddings, *e.g.*, fastText (Bojanowski et al., 2017). Although this transformation is computationally efficient, the process is monotonic, weakening the effect of context in representations. As an orthogonal approach, pre-trained masked language models should fit themselves to generate representations of word meaning in context with supervised fine-tuning. However, annotating word meanings in context is non-trivial, and no such resource is abundantly available.

To address these challenges, we propose a method that distils representations of word meaning in context from masked language models via self-supervised learning.[1] Specifically, our model combines the outputs of different hidden layers using a self-attention mechanism (Vaswani et al., 2017). The distillation model is self-supervised using an autoencoder to reconstruct original representations with an automatically generated training corpus. In contrast to the transformation-based approach, our representations preserve useful context information intact.

Experimental results on a range of benchmark tasks show that our representations exhibited a performance competitive with that of the state-of-the-

[1]Code and training corpus are available at https://github.com/yukiar/distil_wic

Findings of the Association for Computational Linguistics: EMNLP 2021, pages 534–546
November 7–11, 2021.

art method that transforms contextualised representations for context-aware lexical semantics. Furthermore, the results confirm that our representations are more effective for composing sentence representations, which contributes to unsupervised STS estimation.

2 Related Work

2.1 Transformation of Word Representations

Previous studies have proposed transformations of contextualised word representations for various purposes. Pooling aggregates multiple representations to perform one of the simplest transformations. Akbik et al. (2019) complement underspecified contexts for named entity recognition, while Bommasani et al. (2020) investigate information captured in layers of pre-trained models. Wang et al. (2019) transform contextualised word representations by inserting them into Skip-gram (Mikolov et al., 2013) to generate static word representations for context-free lexical semantic tasks such as word similarity and analogy prediction.

Transformation has also been used to adjust excessive effects of context that dominate representations. Shi et al. (2019) add a transformation matrix on top of the embedding layer of ELMo (Peters et al., 2018). Their approach derives the matrix such that final representations of the same words in paraphrased sentences become similar, whereas those of non-paraphrases become distant. The study most relevant to the present work was conducted by Liu et al. (2020). They transform the space of word representations towards the rotated space of static word embeddings using a cross-lingual alignment technique (Doval et al., 2018) for context-aware lexical semantic tasks. In principle, these previous studies aim to make contextualised representations less sensitive to contexts through transformation and prevent them from dominating the representations. We adopt an orthogonal approach to derive word in context representations by combining different layers of a pre-trained model while preserving useful context information intact.

2.2 Representation Disentanglement

Disentanglement techniques are relevant to our approach, which generate specialised representations dedicated to a specific aspect. Previous studies typically employed autoencoders, with the encoder learning to disentangle representations and the decoder learning to reconstruct original representations. In style-transfer research, Shen et al. (2017) disentangled content and sentiment, whereas John et al. (2019) and Cheng et al. (2020) disentangled content and style. Apart from style-transfer, Chen et al. (2019) disentangled semantics and syntax to estimate semantic and syntactic similarities between sentences, and Wieting et al. (2020) disentangled language-dependent styles and sentence meanings for STS estimations. The removal of specific attributes from representations is also relevant. Previous studies have proposed methods for removing predetermined attributes instead of disentangling for multi-linguality (Chen et al., 2018; Lample et al., 2018) and debiasing (Zemel et al., 2013; Barrett et al., 2019).

These previous studies assume that disentangled attributes are distinctive, *e.g.*, language-dependent styles and meanings are supposed to be independent of one another. Similarly, studies on attribute removal assume that the removed attributes are independent of the information remaining in the output representations. In contrast, the distillation of word meaning in context requires a subtle balance to the extent that context information is present in the meaning representations. In this study, we design a self-supervision framework to achieve this challenging goal.

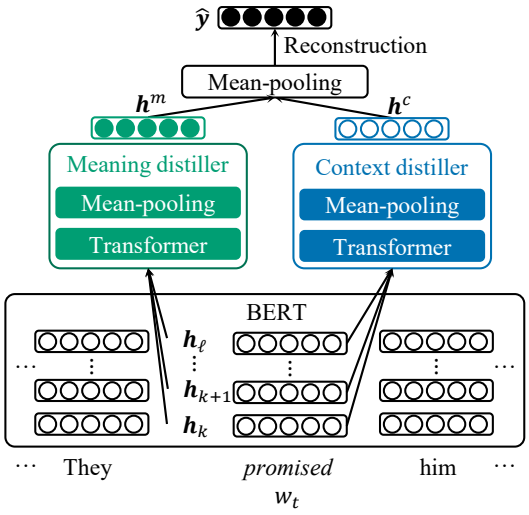

Figure 1: Distillation of word meaning in context via autoencoder

3 Distilling Word Meaning in Context

Inspired by the representation disentanglement approach (Section 2.2), we model the distillation of representations of word meaning in context using an autoencoder framework, as shown in Figure 1.

Vulić et al. (2020) probed pretrained language models for lexical semantic tasks, revealing that lexical information is scattered across lower layers, whereas context-specific information is embedded in higher layers. Hence, we aim to distil the outputs of different hidden layers using a transformer layer. In this study, although we adopted BERT as the masked language model, the proposed method is directly applicable to other pre-trained models.

Figure 1 shows the model architecture. First, we obtain the outputs of all hidden layers of a masked language model, $\mathrm{MLM}(\cdot)$, with frozen parameters $\boldsymbol{H} = \mathrm{MLM}(S) \in \mathbb{R}^{|S| \times (\ell+1) \times d}$, where S is an input sentence of length $|S|$ containing the target word, $w_t \in S$, ℓ is the number of hidden layers in the masked language model (0 corresponding to its embedding layer), and d is the hidden dimension of the masked language model. We then extract the outputs of the hidden layers corresponding to the target word, w_t, from $\boldsymbol{H}$, noting that $\boldsymbol{H}_{w_t} = [\boldsymbol{h}_0, \boldsymbol{h}_1, \cdots, \boldsymbol{h}_\ell]^\intercal \in \mathbb{R}^{(\ell+1) \times d}$. When w_t is segmented into a set of m sub-words $\omega_1, \omega_2, \cdots, \omega_m$, by a tokeniser of the masked language model, we compute the layer-wise averages of the hidden outputs of all sub-words (Bommasani et al., 2020). That is, $\boldsymbol{h}_i \in \boldsymbol{H}_{w_t}$ becomes

$$\boldsymbol{h}_i = \mathrm{Pool}(\boldsymbol{h}_i^{\omega_1}, \cdots, \boldsymbol{h}_i^{\omega_m}),$$

where $\boldsymbol{h}_i^{\omega_j}$ is the ith hidden output of a sub-word ω_j and the $\mathrm{Pool}(\cdot)$ function conducts mean-pooling.

We then input these hidden outputs into a meaning distillation model to derive a representation for word meaning in context. We also input the hidden outputs to another distillation model that derives information other than word meaning in context. For convenience, hereinafter we refer to this information as the *context* and the distillation model as the context distillation model.[2] Each distillation model consists of a transformer layer followed by a mean-pooling function to obtain meaning and context representations, expressed as $\boldsymbol{h}^m \in \mathbb{R}^d$ and $\boldsymbol{h}^c \in \mathbb{R}^d$, respectively.

$$\hat{\boldsymbol{h}}_k, \hat{\boldsymbol{h}}_{k+1}, \cdots, \hat{\boldsymbol{h}}_\ell = \mathrm{TransF}(\boldsymbol{h}_k, \boldsymbol{h}_{k+1}, \cdots, \boldsymbol{h}_\ell),$$

$$\boldsymbol{h}^m = \mathrm{Pool}(\hat{\boldsymbol{h}}_k, \hat{\boldsymbol{h}}_{k+1}, \cdots, \hat{\boldsymbol{h}}_\ell),$$

where $k \in [0, \ell]$ determines the bottom layer to consider and $\mathrm{TransF}(\cdot)$ represents a transformer layer. We distil the context representation in the same manner.

Finally, we reconstruct the original representation from $\boldsymbol{h}^m$ and $\boldsymbol{h}^c$. Although there are different approaches for reconstructions, such as using a neural-network-based decoder, a sophisticated decoder may learn to fit itself to mimic the masked language model outputs. Hence, we adopt mean-pooling as the simplest reconstruction mechanism for reconstruction.

$$\hat{\boldsymbol{y}} = \mathrm{Pool}(\boldsymbol{h}^m, \boldsymbol{h}^c).$$

The reconstruction target $\boldsymbol{y} \in \mathbb{R}^d$ is the mean-pooled hidden layers of the original masked language model.

$$\boldsymbol{y} = \mathrm{Pool}(\boldsymbol{h}_k, \boldsymbol{h}_{k+1}, \cdots, \boldsymbol{h}_\ell). \quad (1)$$

We minimise the reconstruction loss as

$$\mathcal{L}_r = \frac{1}{d} \|\boldsymbol{y} - \hat{\boldsymbol{y}}\|_2^2. \quad (2)$$

For inference, we use $\boldsymbol{h}^m$ as a representation of word meaning in context.

Averaging the outputs of the layers in the top-half of masked language models consistently performs well for context-aware lexical semantic tasks (Vulić et al., 2020; Liu et al., 2020). Thus, we set $k = \ell/2 + 1$ to use the top-half layers for distillation.[3]

John et al. (2019) reported that a variational autoencoder (Kingma and Welling, 2014) outperformed the simpler autoencoder on representation disentanglement. However, this was not the case in this study, wherein the autoencoder consistently outperformed the variational version. We intend to further investigate auto-encoding architectures in future work.

4 Self-supervised Learning

The meaning and context distillation models described in Section 3 require constraints to ensure that the desired attributes are distilled; otherwise, these distillation models obtain a degenerate solution that simply copies the original representations. We design a self-supervision framework ensuring that word meaning in context is distilled using an automatically generated training corpus.

4.1 Cross Reconstruction

Suppose we have two sentences, S_p and S_n. S_p is a sentence that contains a word with the same mean-

[2] The context here should be a mixture of different information that characterises the target word and the sentence, such as the meaning of the entire sentence, syntax, etc.

[3] We also tried $k = 1$ to use all hidden layers, which showed slightly inferior performance to the top-half setting.

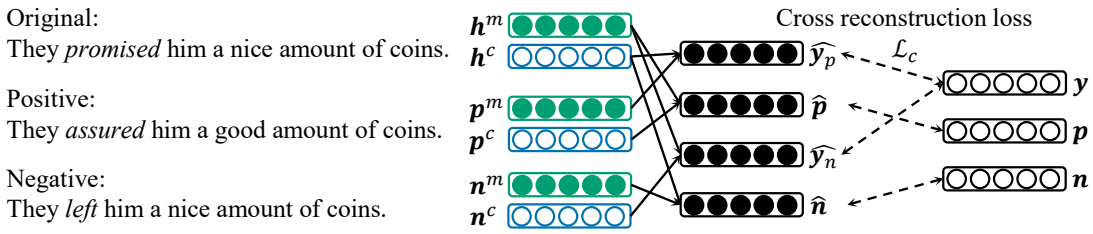

Figure 2: Cross reconstruction with automatically generated positive and negative samples.

original	They *promised* him a nice amount of coins, if the work would be successful.
positive	They *assured* him a good amount of coins if the work was successful.
negative	They *left* him a nice amount of coins, if the work would be successful.

Table 1: Training examples (*Italic* words represent w_t, w_p, and w_n, respectively.)

ing with w_t in S, while S_n contains a word with a different meaning with w_t while the context is the same with S. More concretely, S_p is a sentence containing w_p, which is equivalent to w_t or a lexical paraphrase of w_t, that allows w_p to have the same meaning with w_t in S. In contrast, S_n replaces w_t with a non-paraphrasal word that is suitable for the context, w_n, *i.e.*, $S_n = \{w_n, w_i | w_i \in S \setminus w_t\}$. We refer to S_p and S_n as the positive and negative samples, respectively. Table 1 shows examples of such positive and negative samples.

From the hidden outputs of w_p and w_n, we distil the meaning and context representations, $\boldsymbol{p}^m$ and $\boldsymbol{p}^c$, and those of $\boldsymbol{n}^m$ and $\boldsymbol{n}^c$, respectively. The meaning representation of w_t, $\boldsymbol{h}^m$, should satisfy the following two conditions.

- $\boldsymbol{h}^m$ can be combined with $\boldsymbol{p}^c$ to reconstruct the original representation derived for w_p, and
- $\boldsymbol{h}^m$ can be combined with $\boldsymbol{n}^c$ to reconstruct the original representation, $\boldsymbol{y}$.

Similarly, the context representation, $\boldsymbol{h}^c$, should satisfy the following two conditions.

- $\boldsymbol{h}^c$ can be combined with $\boldsymbol{p}^m$ to reconstruct the original representation, $\boldsymbol{y}$, and
- $\boldsymbol{h}^c$ can be combined with $\boldsymbol{n}^m$ to reconstruct the original representation derived for w_n.

We use these properties of meaning and context representations as constraints.

Specifically, we train the model to achieve cross reconstruction of meaning and context representations, as depicted in Figure 2.

$$\hat{\boldsymbol{p}} = \text{Pool}(\boldsymbol{h}^m, \boldsymbol{p}^c), \quad \hat{\boldsymbol{y}}_p = \text{Pool}(\boldsymbol{p}^m, \boldsymbol{h}^c),$$
$$\hat{\boldsymbol{n}} = \text{Pool}(\boldsymbol{n}^m, \boldsymbol{h}^c), \quad \hat{\boldsymbol{y}}_n = \text{Pool}(\boldsymbol{h}^m, \boldsymbol{n}^c).$$

Our self-supervised learning minimises the following cross reconstruction loss, as given below.

$$\mathcal{L}_c = \frac{1}{d}\{\|\boldsymbol{p} - \hat{\boldsymbol{p}}\|_2^2 + \|\boldsymbol{y} - \hat{\boldsymbol{y}}_p\|_2^2 + \|\boldsymbol{n} - \hat{\boldsymbol{n}}\|_2^2 + \|\boldsymbol{y} - \hat{\boldsymbol{y}}_n\|_2^2\}, \quad (3)$$

where $\boldsymbol{p}$ and $\boldsymbol{n}$ are computed by the same manner with Equation (1). The overall loss function is the summation of the reconstruction and cross-reconstruction losses in Equations (2) and (3)

$$\mathcal{L} = \mathcal{L}_r + \mathcal{L}_c,$$

where $\mathcal{L}_r$ is expanded to sum the reconstruction losses of the positive and negative samples.

4.2 Training Corpus Creation

In this section, we describe the generation of a training corpus for self-supervision using techniques of round-trip translation and masked token prediction.

Round-trip Translation The positive samples in this study require that w_p has the same meaning with w_t in another context of S_p. We assume that common words in a paraphrased sentence pair meet this requirement (Shi et al., 2019). To expand the applicability of our method to various languages, we automatically generate paraphrases using round-trip translation, which translates a source sentence into a target language and then back into the source language. Kajiwara et al. (2020) have shown that pairs of source and back-translated sentences are useful paraphrases for style transfer research. Hence, we obtain S_p by round-trip translation of S.

We need to align w_t and w_p in S and S_p. The two-round translation makes tracing which word

Algorithm 4.1 Simple Word Alignment

Input: Original sentence S containing a target word w_t whose index is t, positive sentence S_p, static word embedding model M, similarity threshold λ

Output: Lexical paraphrase w_p of w_t

1: $M \leftarrow \emptyset, A \leftarrow \emptyset, w_p \leftarrow \emptyset$
2: **for all** $w_i \in S$ and $w_j \in S_p$ **do**
3: $M[i][j] \leftarrow \text{CosineSim}(M(w_i), M(w_j))$ ▷ Compute cosine similarity of embeddings
4: **for all** $w_i \in S \setminus w_t$ **do** ▷ Identify alignments of words other than w_t
5: **if** $j = \text{argmax}\, M[i]$ and $i = \text{argmax}\, M[j]$ **then**
6: $A \leftarrow A \cup \{j\}$
7: **for all** $j \in \text{argsort}(M[t])$ **do** ▷ Sort indices in descending order of $M[t]$
8: **if** $j \notin A$ and $M[t][j] \geq \lambda$ **then**
9: $w_p \leftarrow w_j$
10: break;
11: **return** w_p

in S_p corresponds to w_t non-trivial. Following the trends on monolingual alignment (Yoshinaka et al., 2020) that use static word embeddings, we designed an alignment method based on a simple heuristic using cosine similarities between the embeddings of words in S and S_p, as depicted in Algorithm 4.1. Specifically, we first identify an alignment between word $w_i \in S \setminus w_t$ and $w_j \in S_p$ if and only if they have highest cosine similarities to each other (line 5). We then determine w_p as a word that has the highest cosine similarity to w_t satisfying that it is higher or equal to a pre-determined threshold λ and has not been aligned to others (line 9).

Masked Token Prediction In contrast, negative samples replace w_t with an arbitrary word w_n that fits in the context of S. We generate candidates for replacement words using masked token prediction, which is the primary task used to train the masked language model. Specifically, we input an original sentence whose target is masked by the `[MASK]` label to the masked language model, and we obtain predictions $T = \{t_1, \cdots, t_{|V|}\}$ with probabilities, $Q = \{q_1, \cdots, q_{|V|}\}$, where $|V|$ is the size of the vocabulary of the masked language model. To avoid selecting a possible paraphrase of w_t as w_n, we again use the static word-embedding model following Qiang et al. (2020). We sort T in a descending order of Q and identify w_n the word embedding of which has a lower cosine similarity than λ and a prediction probability q_n higher than a pre-determined threshold δ.

We apply the same technique to enhance w_p when it is identical or similar to w_t based on a character-level edit distance. Where possible, we replace w_p with $w'_p \in T$ the word embedding of which has a higher or equal cosine similarity than λ and a prediction probability higher than δ in masked token prediction.

We also investigated a word substitution approach for self-training corpus creation (Garí Soler and Apidianaki, 2020), *i.e.*, replacing only w_t to w_p using masked token prediction. This method is computationally faster than round-trip translation, but showed inferior performance compared to the proposed approach. We presume this is because round-trip translation provides more diverse lexical paraphrases compared to those already learned by the masked language model, and paraphrasing the context also enhances the robustness of the meaning and context distillers.

5 Experimental Setup

We empirically evaluated whether our method distils representations of word meaning in context from a masked language model using context-aware lexical semantic tasks and STS estimation tasks.[4] All the experiments were conducted on an NVIDIA Tesla V100 GPU.

We compared our method to Liu et al. (2020) as the state-of-the-art in the family of methods that transform contextualised representations. Recall that Liu et al. (2020) adopt an approach orthogonal to that proposed herein, which transforms word representations from the masked language model using static word embeddings. Specifically, we used fastText as the static embeddings that performed most robustly across models and tasks. As a baseline, we also show the performance of BERT. Based on the previous studies (Vulić et al., 2020; Liu et al., 2020), we used the average of the outputs of the top-half layers, *i.e.*, Equation (1), which consistently performed well in lexical semantic tasks.

5.1 Context-aware Lexical Semantic Tasks

We followed experimental settings used by Liu et al. (2020) for a fair and systematic performance com-

[4] We list URLs of all dependent language resources, toolkits, and libraries in the appendix.

LS	# of pairs	STS	# of pairs
USim	1.1k	STS 2012	3.1k
WiC	1.4k	STS 2013	1.5k
CoSimlex-I	680	STS 2014	3.7k
CoSimlex-II		STS 2015	8.5k
SCWS	2k	STS 2016	9.2k

Table 2: Statistics of evaluation corpora

parison. They categorised context-aware lexical semantic tasks into *Within-word* and *Inter-word* tasks. The former evaluates the diversity of word representations for different meanings of the same word associated with different contexts. In contrast, the latter evaluates the similarity of word representations for different words when they have the same meaning. The left-side columns of Table 2 show the number of word pairs in the evaluation corpora.

Within-word Tasks The within-word evaluation was divided into three tasks. The first is based on the Usage Similarity (Usim) corpus (Erk et al., 2013), which provides graded similarity between the meanings of the same word in a pair of different contexts. The second task uses the Word in Context (WiC) corpus (Pilehvar and Camacho-Collados, 2019), which provides binary judgements as to whether the meaning of a given word varies in different contexts. Following the standard setting recommended in the original work, we tuned the threshold for cosine similarity between word representations to make binary judgments. Specifically, we searched the threshold in the range of $[0, 1.0]$ with 0.01 intervals to maximise the accuracy of the development set. The performance of the test set was measured on the CodaLab server.[5] The third task is the subtask-1 of CoSimlex (Armendariz et al., 2020) (denoted as CoSimlex-I). The CoSimlex provides a pair of contexts consisting of a few sentences for each word pair extracted from SimLex-999 (Hill et al., 2015). It annotates the graded similarity in each context. CoSimlex-I requires the estimation of the *change* in similarities between the same word pair in different contexts. Hence, it evaluates whether representations can change for different word meanings according to context.

[5] https://competitions.codalab.org/competitions/20010

Inter-word Tasks The inter-word evaluation consisted of two tasks. The first was the subtask-2 of CoSimlex (denoted as CoSimlex-II), which required estimating the similarity between different word pairs in the same context. The second task used the Stanford Contextual Word Similarity (SCWS) corpus (Huang et al., 2012), which provides graded similarity between word pairs in a pair of different contexts. The contexts of CoSimlex and SCWS consist of several sentences. We input all the sentences as a single context.

Evaluation Metrics We estimated the similarity between words using cosine similarity between their representations. We used evaluation metrics determined by each corpus. Namely, we evaluated WiC using accuracy, CoSimlex-I using Pearson's r, and others using Spearman's ρ.

5.2 STS Tasks

We also evaluated the proposed method on STS tasks. Cosine similarity is commonly used to estimate the similarity between two text representations. In this experiment, we also used cosine similarity because such a primitive measure is sensible to characteristics of different representations. We generated a sentence representation by simply averaging representations of sub-words in a sentence excluding representations for special tokens preserved in BERT, *i.e.*, `[CLS]` and `[SEP]`. We then computed cosine similarities between them.

We evaluated the 2012-to-2016 SemEval STS shared tasks (Agirre et al., 2012, 2013, 2014, 2015, 2016), where the goal is to predict human scores that indicate the degree of semantic similarity between two sentences. The Pearson's r between model predictions and human scores was used as an evaluation metric. Each STS corpus is divided by data sources. Hence, the corpus level score is the average of the Pearson's r for each sub-corpus.

We downloaded and pre-processed STS 2012 to 2016 corpora using the SentEval toolkit (Conneau and Kiela, 2018). The right-side columns of Table 2 show the number of sentence pairs in these corpora.

5.3 Training Corpus Preparation

To prepare a training corpus for self-supervised learning as described in Section 4.2, we used English Wikipedia dumps distributed for the WMT20 competition, the texts of which were extracted using WikiExtractor. As a pre-processing step, we first identified the language of each text using the

langdetect toolkit and discarded all non-English texts. We then conducted sentence segmentation and tokenization using Stanza (Qi et al., 2020) and extracted sentences of 15 to 50 words.

As candidate target words, we extracted the top-50k frequent words[6] following Liu et al. (2020). We then sampled 1M sentences containing these words from the pre-processed Wikipedia corpus. Using these 1M sentences, we generated positive and negative samples via round-trip translation and masked token prediction. For round-trip translation, we trained translators using exactly the same settings as Kajiwara et al. (2020). For convenience, we used fastText as a static word embedding model in Algorithm 4.1. However, other word embeddings or paraphrase lexicons, *e.g.*, PPDB (Ganitkevitch et al., 2013), can also be used. We set λ as 0.6 based on the distribution of cosine similarities of fastText embeddings on a large text corpus.[7] We set δ as 0.003 based on observations of masked token predictions on several samples randomly extracted from the training corpus, such that we could obtain more than 10 predictions of reasonable quality.

Round-trip translation does not always produce an alignable w_p, and our simple word alignment heuristic may fail to identify w_p. Hence, the final number of sentences in our training corpus was reduced to $929,265$, where $44,614$ unique words remained as targets. Among them, $242,643$ sentences had w_p whose surfaces were larger than the 3 character-level edit distance, which were expected as lexical paraphrases. We used these 929k triples of the original, positive, and negative samples for self-supervised learning. We randomly sampled and excluded 10k sentences as a validation set and used the remainder for training.

5.4 Implementation

We implemented our method using PyTorch and Lightning. As a masked language model, we used BERT-Large, cased model for which we used the Transformers library (Wolf et al., 2020). BERT-Large has 24 layers of $1,024$ hidden dimensions with 16 attention heads. Recall that the parameters of BERT were frozen and never fine-tuned.

The meaning and context distillers of the implementation of the proposed model included a transformer layer consisting of $1,024$ hidden dimensions with eight attention heads.[8] We applied 10% dropouts to the transformer layer. The batch size was 128. We used AdamW (Loshchilov and Hutter, 2019) as an optimizer for which the learning rate was tuned as $4.0e-5$ following Smith (2017). For stable training, we applied a warm-up, where the initial learning rate was linearly increased for the first 1k steps to reach the predetermined value. The training was stopped early with a patience of 15 and a minimum delta of $1.0e-4$ based on the validation loss measured for every 0.1 epoch.

For the method of Liu et al. (2020), we replicated their model using the implementation and training corpus published by the authors. Note that their training corpus was also drawn from English Wikipedia. Then, the performance was measured on the same evaluation corpora and computational environments with our method.

6 Results and Discussions

Below, we discuss experimental results and the results of in-depth analyses conducted to identify characteristics of meaning representations generated by our method.

6.1 Experimental Results

Table 3 shows the results on context-aware lexical semantic tasks. The superior performance of our meaning representation to context representations confirm that distillation performed as designed. Our meaning representations achieved performance competitive with the transformation method by Liu et al. (2020).[9] While the transformation method was stronger in Within-Word tasks, our method outperformed it for Inter-Word tasks. This is because the transformation method makes representations of the same words in different contexts closer to the same static embedding but do not explicitly model relations across words. In contrast, our negative samples provide supervision, which makes representations of words with different meanings distinctive. While the performances of these two methods are competitive, these different properties

[6] We excluded the top 0.1% words because most were function words.

[7] This corpus is independent of this study.

[8] We tried 16 attention heads as in the BERT-Large model, but the performance was comparable with that of 8 heads.

[9] The performance of Liu et al. (2020) on CoSimlex-II and SCWS differed from their paper. We suspect the difference was caused by the method used to compose a word-level representation when a word is segmented into sub-words. Because there was no explanation in their paper, we generated the word representation in the same manner with ours, *i.e.*, by layer-wise averaging of all sub-words' hidden outputs (also for the BERT baseline).

		Within-Word			Inter-Word	
		USim (ρ)	WiC (acc.)	CoSimlex-I (r)	CoSimLex-II (ρ)	SCWS (ρ)
BERT-Large		0.5966	66.57	0.7638	0.7332	0.7255
(Liu et al., 2020)		**0.6383**	**67.50**	**0.7710**	0.7258	0.7572
Ours	Meaning	0.6305	67.29	0.7576	**0.7358**	**0.7594**
	Context	0.4147	62.21	0.6485	0.5106	0.2914
w/o NS	Meaning	0.2934	57.79	0.3843	0.5022	0.2883
	Context	0.5929	66.79	0.7617	0.7296	0.7279

Table 3: Results on context-aware lexical-semantic tasks where "w/o NS" denotes the proposed method without negative samples. The best scores are shown in **bold** fonts and scores higher than BERT-Large are underlined (ρ stands for Spearman's ρ, 'acc.' stands for accuracy (%), and r stands for Pearson's r.).

		STS12	STS13	STS14	STS15	STS16
BERT-large		0.480	0.492	0.538	0.589	0.576
(Liu et al., 2020)		0.576	0.616	0.641	0.692	**0.687**
Ours	Meaning	**0.583**	**0.628**	**0.662**	**0.714**	0.684
	Context	0.460	0.411	0.466	0.569	0.575
w/o NS	Meaning	0.177	0.181	0.214	0.238	0.217
	Context	0.573	0.602	0.635	0.706	0.683

Table 4: Results of STS tasks where "w/o NS" denotes our method without negative samples. The best scores are represented in **bold** fonts and scores higher than BERT-Large are underlined.

are reflected in the representations.

This difference is more pronounced in the results of unsupervised STS tasks shown in Table 4. In unsupervised STS tasks, our meaning representations outperformed the transformed representations in four out of five tasks. The transformation has an effect of making contextualised representations less sensitive to contexts to prevent contexts from dominating the representations. This effect is preferred in tasks of context-aware lexical semantics that severely require representations of word meaning, but at the same time, sacrifices context information valuable for other tasks. In contrast, our method does not waste the context information useful for composing sentence representations.

6.2 Analysis

For a deeper understanding of the context information preserved in representations by the transformation method and our method, we conducted an experiment using the corpus of paraphrase adversaries from word scrambling (PAWS) (Zhang et al., 2019). PAWS is a paraphrase corpus dedicated to evaluating the sensitivity of recognition models for syntax in paraphrases. It provides paraphrase and non-paraphrase pairs that were generated by controlled word swapping and back translation with manual screening. Because pairs in PAWS have relatively high word overlap rates, models insensitive to contexts cannot exceed the chance rate for paraphrase recognition.

We generated representations of sentences in the PAWS-Wiki Labeled (Final) section in the same manner as with the STS tasks and computed cosine similarities between them. We then determined a threshold to regard a pair as paraphrase using the development set. Table 5 shows the results. BERT-Large and the transformation method had equal to or lower accuracy than the chance rate of 55.80% (always outputting the majority label of non-paraphrases). In contrast, our method improved the accuracy even on this challenging task. This is achieved by our property that distils word meaning in context without sacrificing useful context information.

6.3 Ablation Study

Table 3 and Table 4 also show results of ablation study, where we left out negative samples for training our method. This left our method uncon-

	Threshold	Accuracy (%)
All False	–	55.80
fastText	1.000	53.89
BERT-Large	0.993	55.76
(Liu et al., 2020)	0.989	55.80
Ours	0.990	**56.71**

Table 5: Paraphrase recognition accuracy on challenging PAWS-Wiki corpus

		Common words		Different words	
		N	P	N	P
Ours	Meaning	0.712	0.754	0.354	0.374
	Context	0.806	0.835	0.580	0.595
w/o NS	Meaning	0.998	0.998	0.996	0.996
	Context	0.705	0.749	0.337	0.357

Table 6: Average cosine similarities between words in PAWS-Wiki where "w/o NS" denotes our method without negative samples ("P" stands for paraphrases and "N" stands for non-paraphrases)

strained; the cross reconstruction became symmetric for the meaning and context distillers. Hence, the model lost its ability to distil word meaning in context into meaning representations. This effect was noticeable for context-aware lexical semantic tasks in Table 3, where meaning representations were no longer useful while the context representations show only a comparable performance to BERT-Large.

Interestingly, these context representations still outperformed representations of BERT-Large on the unsupervised STS tasks. We conducted an intrinsic evaluation again using the PAWS-Wiki Labeled (Final) section to investigate characteristics of the meaning and context representations and reveal possible mechanisms behind this gain. Table 6 shows average cosine similarities between meaning and context representations separately for common and different words in paraphrases and non-paraphrases. Representations for word in context are expected to have (a) higher similarity for words with the same surfaces than for different words, and (b) higher similarity for words appearing in paraphrases than for words in non-paraphrases by reflecting the context. Particularly, appropriate representations should have higher similarity for common words in paraphrases than for those in non-paraphrases because the former more likely has the same meaning.

The meaning and context representations trained with negative samples as well as the context representations without negative samples preserve these characteristics; in other words, they have noticeable distinction between common and different words and words in paraphrases and non-paraphrases. In contrast, the meaning representations generated without negative samples have high cosine similarities among all words, regardless of word and paraphrase relations. This result implies that these meaning representations without negative samples performed as a noise filter to remove non-useful information from the context representations, and only the corresponding context representations benefited from the self-supervision.

7 Summary and Future Work

We have proposed a method that improves contextualised word representations. The proposed approach distils a representation of word meaning in context, retaining useful context information encoded by a masked language model. Experimental results confirmed that our method exhibited performance competitive with the state-of-the-art method for transforming contextualised representations to alleviate excessive effects of contexts on representations, demonstrated on context-aware lexical semantic tasks. Our method further outperformed it on STS tasks.

In a future work, we plan to investigate correspondences of the context representations. We had assumed that these representations preserve the sentence-level meaning; however, the STS results confirmed that this assumption was incorrect. Another possibility is that context representations may retain syntactic information. We intend to conduct in-depth investigations using syntactic tasks. Moreover, we will expand our method to support multilingual masked language models to contribute to cross-lingual processing, *e.g.*, cross-lingual word in context disambiguation (Camacho-Collados et al., 2017), word alignment (Nagata et al., 2020), and quality estimation and post-editing for machine translation (Fomicheva et al., 2020).

Acknowledgments

We appreciate the anonymous reviewers for their insightful comments and suggestions to improve the paper. This project was funded by the Foundation of Kinoshita Memorial Enterprise.

References

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Iñigo Lopez-Gazpio, Montse Maritxalar, Rada Mihalcea, German Rigau, Larraitz Uria, and Janyce Wiebe. 2015. SemEval-2015 task 2: Semantic textual similarity, English, Spanish and pilot on interpretability. In *Proceedings of the International Workshop on Semantic Evaluation (SemEval)*, pages 252–263.

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2014. SemEval-2014 task 10: Multilingual semantic textual similarity. In *Proceedings of the International Workshop on Semantic Evaluation (SemEval)*, pages 81–91.

Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2016. SemEval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In *Proceedings of the International Workshop on Semantic Evaluation (SemEval)*, pages 497–511.

Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. 2012. SemEval-2012 task 6: A pilot on semantic textual similarity. In *Proceedings of the Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 385–393.

Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2013. *SEM 2013 shared task: Semantic textual similarity. In *Proceedings of the Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 32–43.

Alan Akbik, Tanja Bergmann, and Roland Vollgraf. 2019. Pooled contextualized embeddings for named entity recognition. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 724–728.

Carlos Santos Armendariz, Matthew Purver, Matej Ulčar, Senja Pollak, Nikola Ljubešić, and Mark Granroth-Wilding. 2020. CoSimLex: A resource for evaluating graded word similarity in context. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*, pages 5878–5886.

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A simple but tough-to-beat baseline for sentence embeddings. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Maria Barrett, Yova Kementchedjhieva, Yanai Elazar, Desmond Elliott, and Anders Søgaard. 2019. Adversarial removal of demographic attributes revisited. In *Proceedings of Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6330–6335.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association of Computational Linguistics (TACL)*, 5:135–146.

Rishi Bommasani, Kelly Davis, and Claire Cardie. 2020. Interpreting pretrained contextualized representations via reductions to static embeddings. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 4758–4781.

Jose Camacho-Collados, Mohammad Taher Pilehvar, Nigel Collier, and Roberto Navigli. 2017. SemEval-2017 task 2: Multilingual and cross-lingual semantic word similarity. In *Proceedings of the International Workshop on Semantic Evaluation (SemEval)*, pages 15–26.

Mingda Chen, Qingming Tang, Sam Wiseman, and Kevin Gimpel. 2019. A multi-task approach for disentangling syntax and semantics in sentence representations. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 2453–2464.

Xilun Chen, Yu Sun, Ben Athiwaratkun, Claire Cardie, and Kilian Weinberger. 2018. Adversarial deep averaging networks for cross-lingual sentiment classification. *Transactions of the Association of Computational Linguistics (TACL)*, 6:557–570.

Pengyu Cheng, Martin Renqiang Min, Dinghan Shen, Christopher Malon, Yizhe Zhang, Yitong Li, and Lawrence Carin. 2020. Improving disentangled text representation learning with information-theoretic guidance. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 7530–7541.

Alexis Conneau and Douwe Kiela. 2018. SentEval: An evaluation toolkit for universal sentence representations. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*.

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 Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 4171–4186.

Yerai Doval, Jose Camacho-Collados, Luis Espinosa-Anke, and Steven Schockaert. 2018. Improving cross-lingual word embeddings by meeting in the middle. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 294–304.

Katrin Erk, Diana McCarthy, and Nicholas Gaylord. 2013. Measuring word meaning in context. *Computational Linguistics*, 39(3):511–554.

Kawin Ethayarajh. 2018. Unsupervised random walk sentence embeddings: A strong but simple baseline. In *Proceedings of The Third Workshop on Representation Learning for NLP*, pages 91–100.

Kawin Ethayarajh. 2019. How contextual are contextualized word representations? comparing the geometry of BERT, ELMo, and GPT-2 embeddings. In *Proceedings of Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 55–65.

Marina Fomicheva, Shuo Sun, Erick Fonseca, Frédéric Blain, Vishrav Chaudhary, Francisco Guzmán, Nina Lopatina, Lucia Specia, and André F. T. Martins. 2020. MLQE-PE: A multilingual quality estimation and post-editing dataset. *arXiv*, 2010.04480.

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. PPDB: The paraphrase database. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 758–764.

Aina Garí Soler and Marianna Apidianaki. 2020. MULTISEM at SemEval-2020 task 3: Fine-tuning BERT for lexical meaning. In *Proceedings of the International Workshop on Semantic Evaluation (SemEval)*, pages 158–165.

Felix Hill, Roi Reichart, and Anna Korhonen. 2015. SimLex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695.

Eric Huang, Richard Socher, Christopher Manning, and Andrew Ng. 2012. Improving word representations via global context and multiple word prototypes. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 873–882.

Vineet John, Lili Mou, Hareesh Bahuleyan, and Olga Vechtomova. 2019. Disentangled representation learning for non-parallel text style transfer. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 424–434.

Tomoyuki Kajiwara, Biwa Miura, and Yuki Arase. 2020. Monolingual transfer learning via bilingual translators for style-sensitive paraphrase generation. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 8042–8049.

Diederik P. Kingma and Max Welling. 2014. Auto-encoding variational bayes. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Guillaume Lample, Alexis Conneau, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. Word translation without parallel data. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Qianchu Liu, Diana McCarthy, and Anna Korhonen. 2020. Towards better context-aware lexical semantics: Adjusting contextualized representations through static anchors. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4066–4075.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of Conference on Neural Information Processing Systems (NeurIPS)*, volume 26.

Masaaki Nagata, Katsuki Chousa, and Masaaki Nishino. 2020. A supervised word alignment method based on cross-language span prediction using multilingual BERT. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 555–565.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 2227–2237.

Mohammad Taher Pilehvar and Jose Camacho-Collados. 2019. WiC: The word-in-context dataset for evaluating context-sensitive meaning representations. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 1267–1273.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 101–108.

Jipeng Qiang, Yun Li, Zhu Yi, Yunhao Yuan, and Xindong Wu. 2020. Lexical simplification with pretrained encoders. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, page 8649–8656.

Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2017. Style transfer from non-parallel text by cross-alignment. In *Proceedings of Conference on Neural Information Processing Systems (NeurIPS)*, volume 30, pages 6830–6841.

Weijia Shi, Muhao Chen, Pei Zhou, and Kai-Wei Chang. 2019. Retrofitting contextualized word embeddings with paraphrases. In *Proceedings of*

Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 1198–1203.

Leslie N. Smith. 2017. Cyclical learning rates for training neural networks. In *IEEE Winter Conference on Applications of Computer Vision*, pages 464–472.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of Conference on Neural Information Processing Systems (NeurIPS)*, pages 5998–6008.

Ivan Vulić, Edoardo Maria Ponti, Robert Litschko, Goran Glavaš, and Anna Korhonen. 2020. Probing pretrained language models for lexical semantics. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7222–7240.

Yile Wang, Leyang Cui, and Yue Zhang. 2019. How can BERT help lexical semantics tasks? *arXiv*, 1911.02929.

John Wieting, Graham Neubig, and Taylor Berg-Kirkpatrick. 2020. A bilingual generative transformer for semantic sentence embedding. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1581–1594.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 38–45.

Sho Yokoi, Ryo Takahashi, Reina Akama, Jun Suzuki, and Kentaro Inui. 2020. Word rotator's distance. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2944–2960.

Masato Yoshinaka, Tomoyuki Kajiwara, and Yuki Arase. 2020. SAPPHIRE: Simple aligner for phrasal paraphrase with hierarchical representation. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*, pages 6861–6867.

Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. 2013. Learning fair representations. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 325–333.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. PAWS: Paraphrase adversaries from word scrambling. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 1298–1308.

A Dependent Resources

Here is the list of all URLs to the language resources and libraries on which this study depends.

Evaluation corpora

- Usim
  `http://www.dianamccarthy.co.uk/downloads/WordMeaningAnno2012/`
- WiC
  `https://pilehvar.github.io/wic/`
- CoSimlex
  `https://zenodo.org/record/4155986`
- SCWS
  `http://www-nlp.stanford.edu/~ehhuang/SCWS.zip`
- SentEval
  `https://github.com/facebookresearch/SentEval`
- PAWS-Wiki Labeled (Final)
  `https://github.com/google-research-datasets/paws`

Language resources

- English Wikipedia
  `http://data.statmt.org/wmt20/translation-task/ps-km/wikipedia.en.lid_filtered.test_filtered.xz`
- BERT-large, cased
  `https://huggingface.co/bert-large-cased`
- FastText
  `https://dl.fbaipublicfiles.com/fasttext/vectors-english/wiki-news-300d-1M-subword.vec.zip`

Libraries

- WikiExtractor
  `https://github.com/attardi/wikiextractor`
- langdetect
  `https://pypi.org/project/langdetect/`
- Stanza
  `https://stanfordnlp.github.io/stanza/`
- PyTorch (version 1.7.1)
  `https://pytorch.org/`

- Lightning (version 1.1.8)
  https://www.pytorchlightning.ai/
- Transformers (version 4.3.2)
  https://huggingface.co/transformers/
- Implementation of (Liu et al., 2020)
  https://github.com/qianchu/adjust_cwe