Contrastive Learning Approach to Word-in-Context Task for Low-Resource Languages

Pei-Chi Lo, Yang-Yin Lee*, Hsien-Hao Chen*, Agus Trisnajaya Kwee*, Ee-Peng Lim

School of Computing and Information Systems, Singapore Management University
pclo.2017@phdcs.smu.edu.sg

Abstract

Word in context (WiC) task aims to determine whether a target word's occurrences in two sentences share the same sense. In this paper, we propose a Contrastive Learning WiC (CLWiC) framework to improve the learning of sentence/word representations and classification of target word senses in the sentence pair when performing WiC on low-resource languages. In representation learning, CLWiC trains a pre-trained language model's ability to cope with low-resource languages using both unsupervised and supervised contrastive learning. The WiC classifier learning further fine-tunes the language model with WiC classification loss under two classifier architecture options, SGBERT and WiSBERT, which use single-encoder and dual-encoder for encoding a WiC task instance respectively. We evaluate the models developed based on CLWiC framework on a new WiC dataset constructed for Singlish, a low-resource English creole language used in Singapore, as well as the standard English WiC benchmark dataset. Our experiments show that CLWiC-based models using both unsupervised and supervised contrastive learning outperform those not using contrastive learning. This performance difference is more substantial for the Singlish dataset than for the English dataset. Unsupervised contrastive learning appears to improve WiC performance more than supervised one. Finally, we show that using joint learning strategy, we can achieve the best WiC performance.

Introduction

In this paper, our research objective is to address the WiC task for low-resource languages. The Word-in-Context (WiC) task is to determine for a pair of sentences s_1 and s_2, as well as a multi-sense target word w that appears in both s_1 and s_2, if the senses of w triggered by the context in s_1 and s_2 are the same or not. In this binary classification task, the input is a tuple (w, s_1, s_2) and the output is positive if w's occurrences in s_1 and s_2 share the same sense, and negative otherwise. WiC task can be seen as a special case of *word sense disambiguation* (WSD) which is a harder classification task as WSD requires one of the senses (or synsets) from a word sense database to be predicted for a target word occurrence in a sentence (Edmonds and Cotton 2001; Mihalcea, Chklovski, and Kilgarriff 2004). In other words, WSD requires a database covering the word senses of target words. An example of such database is WordNet (Miller 1995)[2], which covers specific senses of a word and each sense is captured by a set of synonyms also known as a *synset*. As such word sense databases are usually missing for low-resource languages, it is appropriate for researchers to first address the WiC task of such languages.

Low-resource languages are those that have little or no data and knowledge resources for training their NLP systems. In this paper, we focus on Singlish a low-resource language widely used in Singapore and Malaysia (Platt 1975). Singlish is essentially a a variant of English heavily influenced by other Asian languages such as Chinese, Malay, Tamil and their dialects. As a result, a Singlish sentence may contain both English and non-English words or phrases. For illustration, the word *shiok* in the Singlish sentence "My food is shiok!" is a malay word that refers to *delicious* or *wonderful*. Despite its popularity in both online and offline conversations among people in the Singapore and Malaysia region, Singlish is not formally taught in schools and an authoritative and up-to-date dictionary does not exist.

WiC for low-resource languages poses several challenges. Firstly, other than the SuperGLUE framework which has been used for rich-resource languages (Wang et al. 2019), there is a lack of other WiC frameworks specifically designed for low-resource languages. Secondly, without sense databases, WiC models that require sense embedding models as part of the solution are no longer applicable (Pelevina et al. 2016; Eyal et al. 2022). Finally, synonyms and homonyms that exist in English also exist in low-resource languages and they complicate the WiC task. For example in Singlish, *jialat* and *chia lat* are different spellings of the same Chinese dialect phrase *chiàh-làt*.

In the following, we summarize our technical contributions:

- We propose a Contrastive Learning WiC framework (CLWiC) as an alternative to the current SuperGLUE framework. CLWiC incorporates different contrastive losses to fine-tune the language model for WiC task and

*This work was done when the author was with Singapore Management University.

Accepted to Workshop on Knowledge Augmented Methods for Natural Language Processing, in conjunction with AAAI 2023.

[2]https://wordnet.princeton.edu

the recognition of synonyms and homonyms, which benefits the low-resource languages. We use Singlish as an example low-resource language in this work.

- For the representation learning module of CLWiC, our contrastive learning approach leverages on implicit senses in the language content and trains the WiC model without using external knowledge resources.
- For the classification module of CLWiC, we propose two architectures, namely SGBERT and WiSBERT, by adding a feature construction layer to the embedding representations generated by the language model.
- To evaluate our CLWiC models, we construct a Singlish WiC dataset and conduct our experiments on both this dataset and the standard English WiC dataset. Our experiments show that CLWiC-based models outperform the baselines, and Singlish appears to benefit even more when using the contrastive learning approach.

Related Works

Word-in-Context Task

Word-in-Context was first defined by Pilehvar and Camacho-Collados (Pilehvar and Camacho-Collados 2019), alongside with a benchmark expert-curated dataset. The WiC benchmark dataset is included as a part of the SuperGLUE benchmark (Wang et al. 2019). The multilingual WiC benchmark, XL-WiC, includes WiC tasks in 12 languages (Raganato et al. 2020). In this paper, we focus on monolingual WiC task.

The previous works addresses WiC in a supervised manner: it learns a classifier trained on the English WiC dataset. Among them, T5 is the state-of-the-art with 76.9% accuracy (Raffel et al. 2020). SemEq aligns definitions of senses from different dictionaries, and learn contextual sense embeddings based on the aggregated definition (Yao et al. 2021). SemEq-Large achieves 75.9% accuracy. ARES which employs semi-supervised approach to generate sense embeddings based on context achieves 72.2% accuracy (Scarlini, Pasini, and Navigli 2020). SenseBERT, which explicitly includes sense information during the training process, reports 72.1% accuracy (Levine et al. 2020). With the external knowledge derived from knowledge bases, KnowBERT achieves 70.9% accuracy on WiC benchmark dataset (Peters et al. 2019). BERT and RoBERTa, which are trained solely on masked language loss without external lexical resources, achieve accuracy of 69.6 and 69.9% respectively (Devlin et al. 2018; Liu et al. 2019).

Contrastive Learning

Contrastive learning (CL) aims to learn a mapping function that generates nearby representations for input data instances that are similar, and far apart representations for disparate data instances. The concept of CL was first introduced by Hadsell et. al as a dimension reduction method (Hadsell, Chopra, and LeCun 2006). Many people have subsequently incorporated CL in representation learning. Most of the works stems from SimCLR, which is a self-supervised CL framework for visual representation learning (Chen et al. 2020). SimCLR employs a stochastic data augmentation module to generate pseudo "positive" samples that are very similar to an original training data instance. It then optimizes a CL loss that attracts the representation of the pseudo positive sample and that of the original training data instance, while repelling the rest of the training data instances. Subsequently, SupCon a supervised CL was proposed (Khosla et al. 2020) to not only attracts training instances that has the same label, but also repels those with different labels. Further, Suresh and Ong proposed weighted supervised CL loss that distinguishes simple negative instances from hard negative instances by assigning higher weight to the latter (Suresh and Ong 2021).

In addition to visual representation learning, previous works have shown that CL helps to generate more robust language models. For instance, COCO-LM jointly learns corrective language modeling which learns to recover the input sentence from a corrupted one, and sequence contrastive learning to bring corrupted sequences originated from the same training sequence closer to one another (Meng et al. 2021). For label-aware CL with language model, PhraseBERT learns to fine-tune BERT to attract phrases and context that are semantically similar while keeping the rest apart (Wang, Thompson, and Iyyer 2021). Designed for sentence paraphrase recognition task (Gao, Yao, and Chen 2021), SimCSE finds that the random dropout masks of transformer-based language models act as simple yet effective data augmentation for unsupervised CL. Finally, PairSupCon jointly optimizes instance discrimination and pairwise semantic reasoning loss to align positive sentence pairs and repels negative pairs (Zhang et al. 2021).

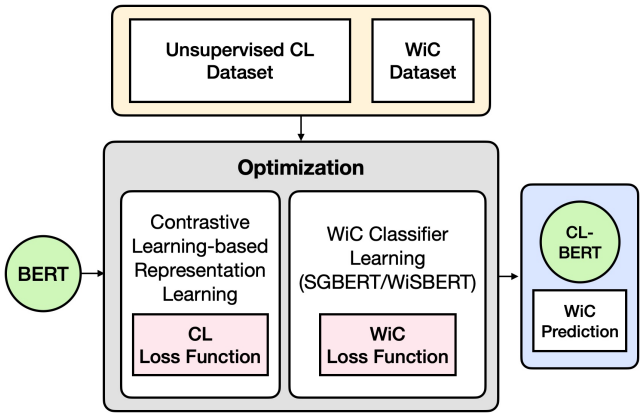

Figure 1: The Contrastive Learning WiC (**CLWiC**) Framework

Proposed CLWiC Framework

CL-based Representation Learning

We show our **Contrastive Learning WiC (CLWiC) framework** in Figure 1. The first module of CLWiC is the CL-based representation learning module which is designed to fine-tune a pre-trained language model such as BERT to generate better-aligned contextual sentence and word embeddings.

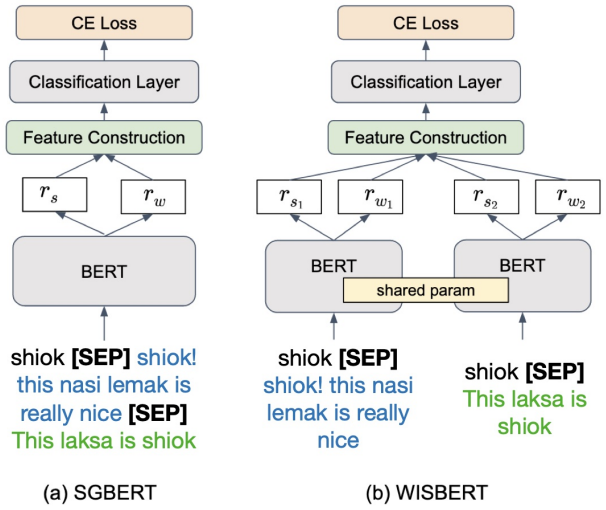

Figure 2: WiC Classifier Learning

We identify two specific goals for CL-based representation learning. Firstly, we want to teach the language model to differentiate the senses of a Singlish word triggered by different sentence contexts. Secondly, we want the language model to generate sentence and word embeddings that yield good WiC prediction accuracy. We introduce *unsupervised contrastive learning* using a very large unlabeled dataset and *supervised contrastive learning* using a small annotated dataset to achieve the first and second goals respectively.

Word-in-Context Classifier Learning

The CLWiC framework includes a classifier that constructs features from the contextual embeddings (e.g., BERT embeddings) for a pair of query sentences and target word, before the features are used to train a WiC classifier. During training, both the BERT model and WiC classifier are fine-tuned and trained respectively to optimize the WiC classification loss (i.e., cross entropy loss). In this project, we propose two different WiC classification architectures, namely **SGBERT** and **WiSBERT**.

SGBERT **SGBERT** architecture is derived from a contextual embedding-based model originally proposed by Yu and Ettinger (Yu and Ettinger 2020) to address paraphrase identification (PI). As shown in Figure 2(a), SGBERT first creates a sequence of tokens for an input tuple (w, s_1, s_2) by having the target word's token(s) appended with word tokens of s_1 and finally with word tokens of s_2. SGBERT then utilises a single BERT encoder to generate the representations r_w, r_{s_1},and r_{s_2} by averaging the BERT's output embeddings of tokens in w, s_1 and s_2 respectively. SGBERT then concatenates these three representations and trains a classifier using representation features. The classifier learns to determine whether (w, s_1, s_2) is a positive or negative tuple by optimizing the cross entropy loss.

WiSBERT While single-encoder is efficient, previous works have found that using a dual-encoder structure to encode sentence separately is flexible and may improve performance. We thus propose the **WiSBERT** model with a dual-encoder structure. As shown in Figure 2(b), WiSBERT first creates two sequences of tokens each involving the target word tokens and a sentence's word tokens. It then generates the representations of the two occurrences of w (denoted by r_{w_1} and r_{w_2}), s_1 (denoted by r_{s_1}) and s_2 (denoted by r_{s_2}). Similar to SGBERT, WiSBERT then learns a classifier using features constructed with r_{s_*}'s and r_{w_*}'s by optimizing the cross entropy loss. While SGBERT combines the target word and sentence representations by concatenation, WiSBERT uses different feature combinations which will be elaborated in the Experiments section.

Learning Strategies

As shown in Figure 1, the CLWiC Framework can perform CL-based representation learning and classifier learning using either the **two-phase learning strategy** or **joint learning strategy**. These two strategies differ in the ways they combine CL-based representation learning loss with WiC classification loss functions in the optimization of model parameters.

Two-Phase Learning Strategy This strategy optimizes the CL-based representation learning and classifier learning as separate steps. In the first phase, we fine-tune an input BERT language model with both unsupervised and supervised contrastive learning losses to obtain an intermediate fine-tuned BERT. In this second phrase, the intermediate fine-tuned BERT is used to generate word and sentence representations for training the WiC classifier. The intermediate BERT is further fine-tuned into CL-BERT as we optimize the WiC classification loss so as to learn the WiC classifier.

Joint Learning Strategy This strategy jointly learns the fully fine-tuned CL-BERT and the WiC classifier with the combined contrastive learning losses and WiC classification loss. Unlike the two-phase learning strategy, it does not produces an intermediate BERT model fine-tuned by contrastive learning only. The joint learning strategy takes a longer training time (about 7hrs in our experiments) but with likely better trained final CL-BERT and WiC classifier.

CL-based Representation Learning

The principle of contrastive learning is to learn better representations by attracting the representations of semantically similar instances to be close to one another and repelling those of semantically unrelated instances to be far from each other (Hadsell, Chopra, and LeCun 2006). In this project, we propose the use of both unsupervised and supervised CL to fine-tune the given BERT model. The unsupervised CL is conducted using a corpus of unlabeled sentences in unsupervised CL dataset as shown in Figure 1. The supervised CL is conducted using the WiC dataset consisting of a set of (w, s_1, s_2) tuples and their labels (positive or negative). A (w, s_1, s_2) tuple is positive if w occurrences in s_1 and s_2 share the same sense, and is negative otherwise. The above two CL methods and their corresponding loss functions are elaborated below.

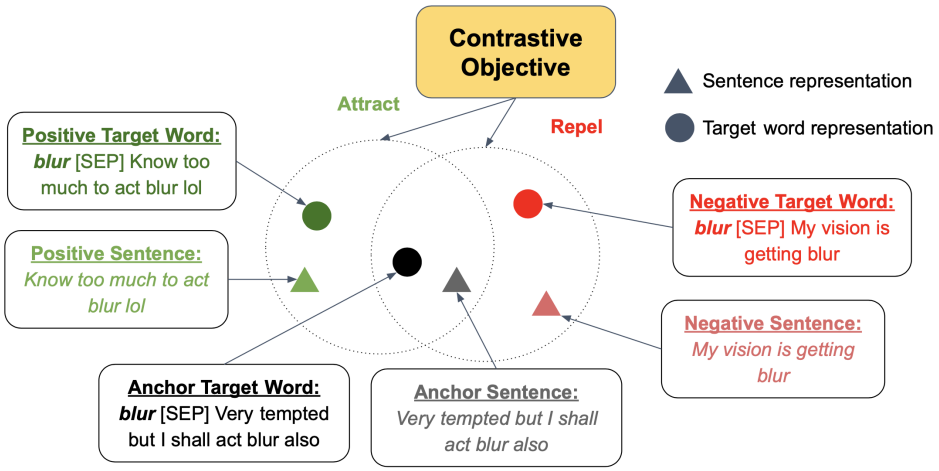

Figure 3: Supervised Contrastive Learning

Unsupervised Contrastive Learning (unCL)

Let $\mathcal{X}$ denote the set of sentences in the unsupervised CL dataset, and s_i be an anchor sentence from a training batch of sentences denoted by $\mathcal{X}_b \subset \mathcal{X}$, and $\mathcal{X}_b = \{s_1, \cdots, s_K\}$. We augment s_i to obtain a positive sample s^+. We also sample negative samples s_j^-'s from the rest of sentences in the training batch $\mathcal{X}_b - \{s_i\}$. Even with data augmentation, s^+ is expected to be still similar to s_i compared to the negative samples $s \in \mathcal{X}_b - \{s_i\}$. Unsupervised CL therefore fine-tunes the BERT model to bring positive sample s^+ closer to s_i and negative samples farther away.

Data Augmentation To generate s^+, we may apply different data augmentation methods including input-level augmentation (e.g., token masking, token deleting, token shuffling, and others) and latent-level transformation (Bhattacharjee, Karami, and Liu 2022). In this project, we adopt the method proposed in SimCSE (Gao, Yao, and Chen 2021) which applies random dropout mask of the language model (say, BERT) to introduce some noise to the sentence representation of s_i. When passing the anchor sentence s_i to the pre-trained BERT model twice each with a different dropout mask, two sentence representations are obtained. The first sentence representation is used as the anchor sentence representation and the second one is regarded as the representation of the augmented sentence s^+, which actually does not have its sentence form. This process ensures that s_i has a representation different from that of s^+ but still keeping most of the anchor's semantics.

Loss Function of Unsupervised Contrastive Learning Let z_i and z_i' be the dropout masks randomly sampled for anchor sentence s_i and its augmented sentence s_i^+. We denote the representations of s_i and s_1' by $h_i = f(s_i, z_i)$, and $h_i' = f(s_i, z_i')$ respectively. The unsupervised contrastive learning strategy fine-tunes the language model $f(\cdot)$ by optimizing the following unsupervised CL loss: $\mathcal{L}_{sent}^{\text{unCL}} = -\sum_{s_i \in \mathcal{X}_b} \log \frac{\exp(h_i \cdot h_i'/\tau)}{\sum_{s_j \in \mathcal{X}_b} \exp(h_i \cdot h_j'/\tau)}$ where τ is the temperature hyper-parameter. Larger τ scales down the dot-product and reduces the emphasis on representation closeness.

In addition to sentence-level unsupervised CL, we propose a word-level unsupervised CL to help generate better contextual word embeddings. From the same collection of sentences, we pick a word from the sentence s_i as w_i. The collection of words from the sentence batch $\mathcal{X}$ is thus denoted by $\mathcal{X}_b^w = \{w_{i1}, \cdots, w_{iK}\}$. We can also introduce another unsupervised contrastive loss function $\mathcal{L}_{word}^{\text{unCL}}$ by replacing sentence representations in the above equation by that of words.

Supervised Contrastive Learning (sCL)

Unlike unsupervised CL, supervised CL focuses on adapting a language model to WiC classification. Here, we propose *word-word* and *word-sentence supervised CL options*, each with a different loss function. We do not include the *sentence-sentence supervised CL* option because the meanings of different sentences with the same target word sense can still be very distinct.

Word-Word Supervised CL Under the word-word option, we extract a set of triplets $(\mathbf{w}_1, \mathbf{w}_1^+, \mathbf{w}_1^-)$'s from the tuples of WiC dataset. Each triplet $(\mathbf{w}_1, \mathbf{w}_1^+, \mathbf{w}_1^-)$ is extracted from a pair of WiC tuples (w, s_i, s_j) and $(w, s_i, s_{j'})$ which share the same target word w and anchor sentence s_i. Moreover, (w, s_i, s_j) and $(w, s_i, s_{j'})$ are assigned with a positive label and negative one respectively. We call the w's occurrence in s_i the *anchor word* and denote it by $\mathbf{w}_1$. The occurrences of w in s_j and $s_{j'}$ are called the *positive word* and *negative word* respectively, and are denoted by $\mathbf{w}_1^+$ and $\mathbf{w}_1^-$ respectively. In word-word supervised CL, we want to bring the representation of $\mathbf{w}_1^+$ closer to that of $\mathbf{w}_1$, and push the representation of $\mathbf{w}_1^-$ away from $\mathbf{w}_1$.

Word-word supervised CL can be shown in Figure 3 which depicts an example triplet with an anchor word (in black circle) and its positive word (in green circle) and negative word (in red circle). By bringing the representations of

anchor and positive words together and keeping that of anchor and negative words apart, supervised CL guides the language model to generate contextualised representations considering word senses. In other words, same-sense words(i.e., synonyms) will be brought together in the vector space and others will be kept away from each other.

Word-Sentence Supervised CL Under the word-sentence option, we also extract a set of triplets $(\mathbf{w_l}, \mathbf{s_l^+}, \mathbf{s_l^-})$'s from the tuples of WiC dataset. Each triplet $(\mathbf{w_l}, \mathbf{s_l^+}, \mathbf{s_l^-})$ is extracted from a pair of WiC tuples (w, s_i, s_j) and $(w, s_i, s_{j'})$ similar to that for word-word supervised CL. The w's occurrence in s_i is then the *anchor word* and denoted by $\mathbf{w_l}$. The positive sentence $\mathbf{s_l^+}$ and negative sentence $\mathbf{s_l^-}$ of the anchor word are s_j and $s_{j'}$ respectively. In word-sentence supervised CL, we want to bring the representation of $\mathbf{s_l^+}$ closer to that of $\mathbf{w_l}$, and push the representation of $\mathbf{s_l^-}$ away from that of $\mathbf{w_l}$ as shown in Figure 3. Word-sentence supervised CL is also illustrated in the lower part of Figure 3 where the green and red triangles denote the representations of positive and negative sentences respectively.

Loss Function of Supervised Contrastive Learning For word-word supervised CL using a collection of word triplets $\mathcal{X}_b^{tp} = \{(\mathbf{w_1}, \mathbf{w}_1^+, \mathbf{w}_1^-), \cdots, (\mathbf{w}_K, \mathbf{w}_K^+, \mathbf{w}_K^-)\}$. Let $h_l = f(\mathbf{w_l})$ denote the language model f that generates the representation h_l of a word $\mathbf{w}_l$. Let $h_l^+ = f(\mathbf{w}_l^+)$ and $h_l^- = f(\mathbf{w}_l^-)$ denote the representations of $\mathbf{w}_l^+$ and $\mathbf{w}_l^-$ respectively. The supervised contrastive learning fine-tunes the language model by optimizing the following loss: $\mathcal{L}_{w2w}^{\text{sCL}} = -\sum_{1\le l\le K} \log \frac{\exp(h_l \cdot h_l^+/\tau)}{\sum_{j=1}^{K}(\exp(h_l \cdot h_j^+)/\tau + \exp(h_l \cdot h_j^-)/\tau)}$

For word-sentence supervised CL loss $\mathcal{L}_{w2s}^{\text{sCL}}$, we replace the h_l, h_l^+, h_j^+ and h_j^- representations in the above equation by that of the anchor word, its positive sentence, other positive sentence and other negative sentence respectively.

Experiments

Datasets

Through experiments, we aim to determine if WiC performance can benefit from contrastive learning, especially for Singlish compared with English. For unsupervised contrastive learning, we use the two unsupervised CL datasets for Singlish WiC (sgWiC) and English WiC (enWiC) respectively.

- **sgWiC unsupervised CL dataset.** For sgWiC, we sampled one million tweets posted by Singapore users in 2020. This dataset, denoted by **sgTweets**, is used for sentence-level unsupervised CL. We then determined Singlish words in a subset of sgTweet sentences and constructed the **sgTweets+SW** dataset. We obtained 129,525 (Singlish word, tweet sentence) pairs for word-level unsupervised CL.
- **enWiC unsupservised CL dataset.** For enWiC, we use the 1M **Wiki** CL dataset provided by the SimCSE work (Gao, Yao, and Chen 2021). We used this dataset for sentence-level unsupervised CL. From the Wiki Dataset, we randomly selected 130,000 sentences and sampled one word from each sentence as the target English word for word-level unsupervised CL. This forms the **Wiki+TG** dataset.

Table 1: Dataset Statistics

Unsupervised CL Datasets		**Total**
sgWiC	sgTweets	1,000,000
	sgTweets + SW	129,525
enWiC	Wiki	1,000,000
	Wiki + TG	130,000

Supervised CL Datasets	**#w2w SCL Tri.**	**#w2s sCL Tri.**
sgWiC	4,163	8,326
enWiC	6,066	12,132

WiC Datasets	**#Train**	**#Dev**	**#Test**	**Total**
sgWiC	3,500	388	275	4,163
enWiC	5,228	638	200	6,066

We also include two WiC datasets for evaluating the sgWiC and enWiC performance or supervised constrastive learning. We show the statistics of the datasets in Table 1.

- **Singlish-WiC Dataset (sgWiC).** This WiC dataset is a collection of (w, s_i, s_j) tuples such that w is a Singlish word, and s_i and s_j are Singaporean tweet sentences that contain w. The WiC labels of these tuples are crowdsourced from Singapore users. We collected a dataset of 4,163 tuples (2,081 positives and 2,082 negatives) and split them into training, development, and testing data. From these tuples, we construct 2,082 word-word-word triplets for word-word supervised CL training, and another 2,082 word-sentence-sentence triplets for word-sentence supervised CL training.
- **English-WiC Benchmark Dataset (enWiC).** This dataset consists of WiC data in English context and has been downloaded from **WiC:The Word-in-Context Dataset** published in NAACL'19 [3] (Pilehvar and Camacho-Collados 2019). This downloaded version of WiC data covers the training data only for the WiC competition [4]. We thus sample 200 instances (100 positives and 100 negatives) from the training set as our testing data. The training data consists of 2,614 positives and 2,614 negatives. In total, there are 1,855 distinct words in this dataset.

Model Comparison

In our experiment, we evaluate the performance of different models developed based on the CLWiC framework. To distinguish these different CLWiC-based models, we introduce the following model naming convention: **CLWiC(⟨classification architecture⟩, ⟨contrastive learning⟩, ⟨learning strategy⟩)**. Each model is identified by its classification architecture, contrastive learning, learning strategy and feature construction options elaborated below.

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

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

Classification architecture options Two classification architectures mentioned in Section are:

- **SB** (or SGBERT): For SGBERT architecture, we take the word and sentence representations from the language model r_w, r_{s_1}, and r_{s_2}, and concatenate them to form the feature vector $x = [r_w, r_{s_1}, r_{s_2}]$.
- **WB** (or WiSBERT): For WiSBERT architecture, we include a few different feature construction sub-options to derive the output feature vector x from the word and sentence representations $(r_{s_1}, r_{w_1}, r_{s_2}, r_{w_2})$ returned by the language model as shown in Figure 2. The feature construction sub-options are:

  WB[BAS] (Basic): $x_{\text{BAS}} = [r_s^1, r_s^2, r_w^1, r_w^2]$
  WB[HAD] (Hadamard Product): $x_{\text{HAD}} = [r_s^1 \odot r_s^2, r_w^1 \odot r_w^2]$
  WB[DIFF] (Differentiation): $x_{\text{DIFF}} = [r_s^1 - r_s^2, r_w^1 - r_w^2]$
  WB[COS] (Cosine Similarity): $x_{\text{COS}} = [cos(r_s^1, r_s^2), cos(r_w^1 \cdot r_w^2)]$, where $cos(\cdot)$ is the cosine similarity measurement. Early experiment results suggest that using cosine similarity features alone leads to ill-optimization of the language model. As a result, this feature will only be used together with other features.
  WB[ALL] (All): $x_{\text{ALL}} = [x_{\text{BAS}}, x_{\text{HAD}}, x_{\text{DIFF}}, x_{\text{COS}}]$

For English WiC task, we use BERT-based-uncased[5] for enWiC. For Singlish WiC task, we use SingBERT[6] and further fine-tuned it on 673,205 Singlish tweets as our base language model for sgWiC..

Contrastive learning options The following contrastive learning (CL) options can be used:

- **noCL** (Without using any contrastive learning loss): In this case, only the WiC classification loss $\mathcal{L}^{\text{WiC}}$ is used in model training.
- **unCL** (Unsupervised contrastive learning only): In addition to WiC loss, we fine-tune the BERT model with unsupervised contrastive loss $\mathcal{L}^{\text{unCL}}$. Three unsupervised CL loss sub-options are available:

  unCL[sent]: Sentence level unCL, i.e., $\mathcal{L}^{\text{unCL}} = \mathcal{L}^{\text{unCL}}_{sent}$)
  unCL[word]: Word level unCL, i.e., $\mathcal{L}^{\text{unCL}} = \mathcal{L}^{\text{unCL}}_{word}$)
  unCL[full]: Both sentence and word level unCL, i.e., $\mathcal{L}^{\text{unCL}} = \mathcal{L}^{\text{unCL}}_{sent} + \mathcal{L}^{\text{unCL}}_{word}$.
- **sCL** (Supervised Contrastive Learning only): In addition to WiC loss, we apply supervised contrastive loss $\mathcal{L}^{\text{sCL}}$ to fine-tune the model. There are three sCL options, namely:

  sCL[w2w]: Word-to-word sCL loss, i.e., $\mathcal{L}^{\text{sCL}} = \mathcal{L}^{\text{sCL}}_{w2w}$)
  sCL[w2s]: Word-to-sentence sCL loss, i.e., $\mathcal{L}^{\text{sCL}} = \mathcal{L}^{\text{sCL}}_{w2s}$)
  sCL[full]: Full supervised CL loss, i.e., $\mathcal{L}^{\text{sCL}} = \mathcal{L}^{\text{sCL}}_{w2w} + \mathcal{L}^{\text{sCL}}_{w2s}$.
- **fullCL** (Full contrastive learning): We fine-tune the BERT model using both full unsupervised CL loss, $\mathcal{L}^{\text{unCL}} = \mathcal{L}^{\text{unCL}}_{sent} + \mathcal{L}^{\text{unCL}}_{word}$, and full supervised CL loss, $\mathcal{L}^{\text{sCL}} = \mathcal{L}^{\text{sCL}}_{w2w} + \mathcal{L}^{\text{sCL}}_{w2s}$.

[5]https://huggingface.co/bert-base-uncased

[6]https://huggingface.co/zanelim/singbert

Learning Strategy options We compare the two learning strategy options mentioned in Section :

- **2PL** (Two-Phase Learning Strategy): This strategy has been explained in Section . We include two fine-tuning sub-options for training of language model in Phase 2 (WiC classifier training), namely:

  2PL[static]: Here, we do not fine-tune the language model further in Phase 2 with the WiC classification loss $\mathcal{L}^{\text{WiC}}$ in Phase 2.
  2PL[fine-tuning]: For this sub-option, the language model is fine-tuned in Phase 2 with the WiC classification loss function.

- **JL** (Joint Learning Strategy): We fine-tune the language model and train the WiC classifier jointly by optimizing both the contrastive loss and WiC classification, i.e., $\mathcal{L} = \lambda(\mathcal{L}^{\text{CL}}) + (1-\lambda)\mathcal{L}^{\text{WiC}}$. λ is a hyperparameter that controls how much the CL loss contributes to the overall optimization. We empirically use $\lambda = 0.5$ in the experiments.

Based on the above model naming convention, the SuperGLUE model is equivalent to the **CLWiC(SB,noCL,2PL[static])** model. Other non-CL based models are **CLWiC(WB[·],noCL,·)**'s.

Results and Discussion

Tables 2 and 3 show the word-in-context accuracy for the sgWiC and enWiC datasets respectively. The overall best performance is **boldfaced**, and the best performance without CL is underlined.

We first make several observations from the sgWiC experiment results in Table 2. The best performing model is **CLWiC(WB[ALL], fullCL, JL)** with 70.2% accuracy, followed by 68.5% from **CLWiC(WB[DIFF], fullCL, JL)**. For Singlish WiC task, it is clear that SuperGLUE framework does not perform as well as our proposed CLWiC framework. WiSBERT classification architecture using all features, full contrastive learning and joint learning strategy all contribute well to the sgWiC task. **CLWiC(WB[ALL], fullCL, JL)** is also far better than the 56% accuracy of state-of-the-art SuperGLUE model **CLWiC(SB, noCL, 2PL[static])** which could not cope with sgWiC dataset well.

Contrastive learning using unsupervised CL with unCL[full] option, or supervised CL with sCL[full] option generally outperforms the methods not using contrastive learning. When combining both unsupervised and supervised CL together (i.e., fullCL option), the improvement is usually substantial. For example, **CLWiC(WB[ALL], fullCL, 2PL[FT])** achieves 12% improvement in accuracy over non-contrastive learning method using WiSBERT with ALL feature option.

Among the models using full contrastive learning (i.e., fullCL), those using the joint learning option is generally better than those using 2-phase learning. Between unCL and sCL options, unCL appears to contribute more to the accuracy improvement. This could be due to the availability of large datasets for unsupervised CL. Under unsupervised CL, it is however unclear whether sentence- or word-level unCL

Table 2: Performance on sgWiC Dataset (Accuracy)

		noCL	CL using 2PL							CL using JL		
			2PL	unCL			sCL			full	unCL	sCL	full
			[*]	sent	word	full	w2w	w2s	full	CL	full	full	CL
WiSBERT	ALL	0.575	FT	0.629	0.592	0.631	0.615	0.581	0.613	0.645	0.671	0.625	**0.702**‡
	BAS	0.622		0.593	0.620	0.625	0.556	0.577	0.587	0.630	0.629	0.604	0.630
	DIFF	0.629		0.582	0.607	0.614	0.615	0.604	0.632	0.629	0.647	0.634	0.685
	HAD	0.509		0.599	0.601	0.612	0.524	0.548	0.566	0.615	0.614	0.572	0.613
WiSBERT	ALL	0.549	ST	0.592	0.587	0.587	0.567	0.555	0.570	0.591	Not Applicable†		
	BAS	0.549		0.596	0.573	0.596	0.527	0.523	0.532	0.562			
	DIFF	0.535		0.593	0.601	0.602	0.527	0.531	0.535	0.584			
	HAD	0.567		0.623	0.633	0.627	0.573	0.569	0.583	0.627			
SGBERT		0.575	**FT**	0.622	0.613	0.625	0.567	0.571	0.577	0.613	0.629	0.584	0.643
SGBERT		0.560	**ST**	0.585	0.577	0.588	0.578	0.575	0.578	0.591	Not Applicable		

* 2PL[FT] and 2PL[ST] refers to BERT model fine-tuning and static sub-options respectively for 2-Phase Learning strategy.
† BERT model is always fine-tuned under joint learning strategy.

Table 3: Performance on English WiC Dataset (Accuracy)

		noCL	CL using 2PL							CL using JL		
			2PL	unCL			sCL			full	unCL	sCL	full
			[*]	sent	word	full	w2w	w2s	full	CL	full	full	CL
WiSBERT	ALL	0.645	FT	0.660	0.650	0.680	0.645	0.645	0.675	0.695	0.695	0.680	0.705
	BAS	0.630		0.630	0.630	0.630	0.635	0.625	0.625	0.660	0.645	0.630	0.670
	DIFF	0.645		0.660	0.650	0.660	0.650	0.650	0.645	0.685	0.670	0.645	0.690
	HAD	0.600		0.600	0.605	0.605	0.595	0.590	0.595	0.610	0.615	0.595	0.620
SGBERT		0.735	**FT**	0.740	0.740	0.750	0.725	0.720	0.735	0.760	0.750	0.740	**0.775**

is better. Similarly, we do not see any clear performance difference between w2w and w2s options under supervised CL.

Models using 2-Phase learning with fine-tuning consistently outperform their corresponding static ones. For example, **CLWiC(WB[ALL], fullCL, 2PL[fine-tuning])** outperforms **CLWiC(WB[ALL], fullCL, 2PL[static])**. We therefore leave out the latter for the English WiC experiments.

Finally, For feature comparison, we focus on models using WiSBERT with full contrastive learning and joint learning strategy. The best feature option is ALL followed by DIFF. For static WiC models, HAD is the best feature option followed by ALL. We believe this inconsistent observation could be attributed to the small sized sgWiC dataset.

As shown in Table 3, for English WiC, we conclude that the best performing model is **CLWiC(SB, fullCL, JL)** with 77.5% accuracy. This model also outperforms the SuperGLUE model denoted by **CLWiC(SB, noCL, 2PL[static])** which has an accuracy of 73.5%. This result shows that contrastive learning also effectively improves the results of enWiC task. Interestingly, SGBERT outperforms WiSBERT across all model options. The opposite finding was observed in the sgWiC results. The better performance of SGBERT could be explained by the enWiC task being easier than sgWiC. For example in sgWiC, the same Singlish word could have different spellings. Again, models using unsupervised CL outperforms those using supervised CL models as observed in Singlish results.

Conclusion and Future Works

In this paper, we propose a contrastive learning WiC (CLWiC) framework to address Word-in-Context task involving low-resource languages. We propose both unsupervised (unCL) and supervised contrastive learning (sCL) to fine-tune pre-trained language models (e.g., BERT) for matching word senses. Our experiments on both Singlish and English WiC datasets shows the contrastive learning based models with both unCL and sCL losses, trained together with WiC classification loss, outperform the existing baseline model. These results suggest that contrastive learning can help to improve WiC task for both low resource and rich resource languages. Our results demonstrate that even unlabeled low-resource language data can be used in unsupervised contrastive learning to achieve better WiC results.

To further improve the WiC performance, one future research direction is to improve the generalisability of these CL-based models. With new words and phrases emerging in low-resource languages, it is important to study how these models can be extended to handle WiC tasks with unseen words. As models trained on larger and more diverse datasets often have better generalisability, we plan to manually annotate more datasets as well as to construct semi- or fully-automated annotated datasets to develop more accurate CL-based WiC models for low-resource languages.

References

Bhattacharjee, A.; Karami, M.; and Liu, H. 2022. Text Transformations in Contrastive Self-Supervised Learning: A Review. In *IJCAI*.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020. A simple framework for contrastive learning of visual representations. In *ICML*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Edmonds, P.; and Cotton, S. 2001. SENSEVAL-2: Overview. In *SENSEVAL*.

Eyal, M.; Sadde, S.; Taub-Tabib, H.; and Goldberg, Y. 2022. Large Scale Substitution-based Word Sense Induction. In *ACL*, 4738–4752.

Gao, T.; Yao, X.; and Chen, D. 2021. SimCSE: Simple Contrastive Learning of Sentence Embeddings. In *EMNLP*.

Hadsell, R.; Chopra, S.; and LeCun, Y. 2006. Dimensionality reduction by learning an invariant mapping. In *CVPR*.

Khosla, P.; Teterwak, P.; Wang, C.; Sarna, A.; Tian, Y.; Isola, P.; Maschinot, A.; Liu, C.; and Krishnan, D. 2020. Supervised Contrastive Learning. In *NeurIPs*, volume 33, 18661–18673.

Levine, Y.; Lenz, B.; Dagan, O.; Ram, O.; Padnos, D.; Sharir, O.; Shalev-Shwartz, S.; Shashua, A.; and Shoham, Y. 2020. SenseBERT: Driving Some Sense into BERT. In *ACL*.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv preprint arXiv:1907.11692*.

Meng, Y.; Xiong, C.; Bajaj, P.; tiwary, s.; Bennett, P.; Han, J.; and SONG, X. 2021. COCO-LM: Correcting and Contrasting Text Sequences for Language Model Pretraining. In *NeurIPs*.

Mihalcea, R.; Chklovski, T.; and Kilgarriff, A. 2004. The Senseval-3 English lexical sample task. In *SENSEVAL*.

Miller, G. A. 1995. WordNet: a lexical database for English. *Communications of the ACM*, 38(11): 39–41.

Pelevina, M.; Arefiev, N.; Biemann, C.; and Panchenko, A. 2016. Making Sense of Word Embeddings. In *Proceedings of the 1st Workshop on Representation Learning for NLP*.

Peters, M. E.; Neumann, M.; Logan, R.; Schwartz, R.; Joshi, V.; Singh, S.; and Smith, N. A. 2019. Knowledge Enhanced Contextual Word Representations. In *EMNLP-IJCNLP*.

Pilehvar, M. T.; and Camacho-Collados, J. 2019. WiC: the Word-in-Context Dataset for Evaluating Context-Sensitive Meaning Representations. In *NAACL*.

Platt, J. T. 1975. The Singapore English speech continuum and its basilect'Singlish'as a'creoloid'. *Anthropological linguistics*, 363–374.

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*, 21(140): 1–67.

Raganato, A.; Pasini, T.; Camacho-Collados, J.; and Pilehvar, M. T. 2020. XL-WiC: A Multilingual Benchmark for Evaluating Semantic Contextualization. In *EMNLP*.

Scarlini, B.; Pasini, T.; and Navigli, R. 2020. With More Contexts Comes Better Performance: Contextualized Sense Embeddings for All-Round Word Sense Disambiguation. In *EMNLP*.

Suresh, V.; and Ong, D. 2021. Not All Negatives are Equal: Label-Aware Contrastive Loss for Fine-grained Text Classification. In *EMNLP*.

Wang, A.; Pruksachatkun, Y.; Nangia, N.; Singh, A.; Michael, J.; Hill, F.; Levy, O.; and Bowman, S. 2019. SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems. In Wallach, H.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E.; and Garnett, R., eds., *NeurIPs*.

Wang, S.; Thompson, L.; and Iyyer, M. 2021. Phrase-BERT: Improved Phrase Embeddings from BERT with an Application to Corpus Exploration. In *EMNLP*.

Yao, W.; Pan, X.; Jin, L.; Chen, J.; Yu, D.; and Yu, D. 2021. Connect-the-Dots: Bridging Semantics between Words and Definitions via Aligning Word Sense Inventories. In *EMNLP*. Association for Computational Linguistics.

Yu, L.; and Ettinger, A. 2020. Assessing Phrasal Representation and Composition in Transformers. In *EMNLP*.

Zhang, D.; Li, S.-W.; Xiao, W.; Zhu, H.; Nallapati, R.; Arnold, A. O.; and Xiang, B. 2021. Pairwise Supervised Contrastive Learning of Sentence Representations. In *EMNLP*.