RepAL: A Simple and Plug-and-play Method for Improving Unsupervised Sentence Representations

Anonymous ACL submission

Abstract

Unsupervised sentence representation learning is a fundamental problem in natural language processing and has been studied extensively in recent years. This paper presents **Rep**resentation **AL**chemy (RepAL), an extremely simple post-processing method that enhances unsupervised sentence representations. The basic idea in RepAL is to extract redundant information from the representation of a sentence generated by the existing models and then refine the representation through an embedding refinement operation to filter such redundant information. In this paper, we analyze the redundant information from two levels: sentence-level and corpus-level, and the theoretical analysis for the latter is also conducted. We point out that RepAL is free of training and is a plug-and-play method that can be combined with most existing unsupervised sentence learning models. Extensive experiments demonstrate RepAL's effectiveness and show that RepAL is a model-agnostic method for unsupervised sentence embedding enhancement. Besides, we also designed detailed ablation studies to understand why RepAL works and provided in-depth analysis and understanding of the redundant information.

1 Introduction

Learning high-quality sentence embeddings is a fundamental task in Natural Language Processing (NLP) field (Socher et al., 2011; Le and Mikolov, 2014; Kiros et al., 2015; Reimers and Gurevych, 2019; Gao et al., 2021). The goal is to map semantically similar sentences close and dissimilar sentences farther apart in the representation space. In real-world scenarios, especially when a large amount of supervised data is unavailable, an approach that provides sentence embeddings in an unsupervised paradigm is of great value.

Generally, the unsupervised sentence encoder (USE) can be categorized into two paradigms. The first is pre-trained language model (PTM) (Devlin et al., 2019; Liu et al., 2019) based paradigm, which are naturally good unsupervised sentence representation learning models. For example, BERT (Devlin et al., 2019) and BERT-like (Liu et al., 2019; He et al., 2020; Raffel et al., 2020) models, commit to design stronger pre-trained language models by self-training with mask or next sentence prediction. While designing stronger PTMs for is extremely expensive, time-consuming, and labor-intensive. Based on PTMs, secondary trained, e.g., contrastive-based methods (Reimers and Gurevych, 2019; Logeswaran and Lee, 2018; Gao et al., 2021), proved to be effective to further improve the representation quality of sentences. As a representative solution, SimCSE (Gao et al., 2021) minimizes the distance between positive pairs of sentences and pulls away from the negative pairs of sentences in the embedding space, which is highly dependent on a crucial data augmentation trick to create positive sentence pair. Nevertheless, designing NLP augmentation tricks that significantly outperform dropout (Srivastava et al., 2014) remains challenging.

This paper argues that the quality of sentence embeddings generated by most existing unsupervised methods can be further improved with a post-processing operation free of training and extra data. Our basic idea is to *refine sentence representations by removing the redundant information from the levels of sentences and corpus*. Specifically, given a sentence, there are several trivial and inconsequential words within the sentence. Such words are proven to bring a negative impact on downstream NLP tasks, like Natural Language Inference (NLI) (Mahabadi et al., 2020; Zhou and Bansal, 2020) and text classification (Choi et al., 2020; Qian et al., 2021). Besides, given a training corpus, there is some information shared by the whole corpus. Such information may lead to homogenous properties for all sentence embeddings, which

diminishes the distinctiveness between sentences.

This paper proposes a simple, straightforward, and effective method called representation alchemy (RepAL), which improves sentence representations without training and extra resources. RepAL accepts raw sentence representations as inputs, which are generated from existing unsupervised sentence models. Then RepAL outputs refined representations by extracting two redundant representations from different perspectives. Intuitively, it is like *an alchemy that improves sentence representation by impurity refinement*. It's worth mentioning that our proposed RepAL can be applied to almost USEs, and is a plug-and-play method in sentence embedding enhancement.

To verify, we perform extensive experiments on the widely used benchmarks both in English and Chinese. The results demonstrate our RepAL's effectiveness: it can be well combined with existing USE. Besides, we also conduct detailed ablation studies to dive into the capacity of two embedding refinements, respectively. We conclude that sentence-level refinement can diminish the impact of trivial words when measuring semantic similarity. Corpus-level refinement diminishes the largest eigenvector for the embedding matrix, thus improving sentence representation's isotropy (Wang et al., 2019b,a). Our main contributions can be summarized as follows:

- We propose RepAL, a plug-and-play method that enhances the sentence representations through redundant embedding generation and refinement. Experiments show that it can be well combined with different USE and improve their capacity in sentence representations.

- We ablate our method to illustrate the effectiveness of sentence-level and corpus-level refinement operations with theoretical derivation and empirical results.

2 Related Work

Methods for unsupervised sentence learning have been extensively explored. Early works are mainly based on distributional hypothesis (Socher et al., 2011; Le and Mikolov, 2014). Hill (Hill et al., 2016) proposed to learn sentence representations with the internal structure of each sentence. Kiros (Kiros et al., 2015) proposed to predict the surrounding sentences of a given sentence like Word2Vec. Then Pagliardini (Pagliardini et al., 2018) proposed Sent2Vec, a simple unsupervised model allowing to compose sentence embeddings using word vectors along with n-gram embeddings.

Then the strong pre-trained language model (Devlin et al., 2019) emerged from the blue. The powerful pre-trained models own strong potential to improve the quality of sentence representation. However, models like BERT own strong anisotropy in their embedding space which means the sentence embeddings produced by BERT have extremely high cosine similarity, leading to an unsatisfactory performance on sentence embedding.

Some post-processing methods have been proposed to improve the quality of contextual sentence embeddings to solve such a problem. The post-processing paradigm aims to enhance sentence embeddings through simple and efficient operations without extra training or data. The most promising method is whitening (Huang et al., 2021), dedicated to transforming sentence embedding into Gaussian-like embedding, which proved to be effective in sentence embedding improvement.

Recently, contrastive learning began to play an important role in unsupervised sentence representation learning (Zhang et al., 2020; Yan et al., 2021; Meng et al., 2021; Gao et al., 2021; Wang et al., 2021). Such methods are based on the assumption that high-quality embedding methods should bring similar sentences closer while pushing away dissimilar ones. Therefore, those methods use various data augmentation tricks to generate two different views for each sentence and design an effective loss function to make them closer in the representation space. Among the data augmentation tricks, dropout (Srivastava et al., 2014) is one representative and effective method.

Specifically, the most relevant work to ours is whitening (Huang et al., 2021) since the corpus-level refinement is similar to the average embedding subtraction in whitening. However, there are three principal differences between such two works. Firstly, the motivation is different. Whitening aims at transforming the sentence embedding to Gaussian-like embedding for distance measurement on an orthogonal basis. Our method starts a perspective of redundancy refinement, which aims to diminish the impact of trivial words within a sentence during similarity calculation. Second, the methodology is different. Our method additionally employs a partial mask to filter the redundancy

and introduce weight factors to control the impact during embedding refinement. Lastly, the in-depth analysis shows that our method aims to diminish the upper bound of the largest eigenvalue of the embedding matrix and the impact of trivial words, which is irrelevant to whitening's effects.

3 Methodology

3.1 Problem Formulation

In unsupervised sentence representation learning, we take a collection of unlabeled sentences $\{x_i\}_{i=1}^n$, also we choose a suitable unsupervised sentence learning model (e.g., BERT) as the encoder $f(\cdot;\theta)$, where θ represents the trainable parameters in f. Specifically, we have a carefully designed training objective $\mathcal{L}(x_i,\theta)$ for unsupervised training, and θ is then fixed as θ_0 where $\theta_0 = \operatorname{argmin} \mathcal{L}(x_i,\theta)$. Finally, we obtain the sentence representation v_i for x_i by feeding it into the encoder, i.e., $v_i = f(x_i;\theta_0)$.

RepAL plays its role in refining v_i to v_i' with $v_i' = g(v_i)$, instead of directly selecting v_i for sentence representation. RepAL aims to extract and refine two types of redundancy, namely *sentence-level redundancy* and *corpus-level redundancy*, respectively. Sentence-level redundancy denotes the useless word information hidden in the target sentence, which may bias the representations that reflect the core semantics of the sentence. Corpus-level redundancy denotes the shared redundant information in all sentence representations within the dataset, making all the representations tend to be homogenous and thus reducing the distinction.

RepAL generates x_i^* by an operation called *partial mask* on x_i, then feed x_i^* into the encoder $f(.;\theta_0)$ to obtain sentence-level redundancy embeddings v_i^*. Besides, RepAL produces a global vector $\hat{v}$ as corpus-level redundancy embedding. Finally, RepAL generates the refined embedding v_i' for downstream tasks through the *embedding refinement* operation by combining v_i, v_i^* and $\hat{v}$. The overview architecture of RepAL is illustrated in Figure 1, which consists of two principal stages: redundant embedding generation and embedding refinement.

3.2 Redundant Embedding Generation

In RepAL, we firstly detect redundant information and generate their embeddings from the target sentence, which is a groundbreaking step in our method and determines the performance.

3.2.1 Sentence-level Redundancy

We apply a partial mask to extract the sentence-level redundancy. Specifically, given a sentence $x_i = \{w_1, w_2, \ldots, w_N\}$, partial mask generates a partially masked sentence x_i^*, a mask version of x_i, where *informative* words in x_i are replaced with [MASK] to distill the trivial words from the sentence. Specifically, we judge the words as keywords according to their TF-IDF (Luhn, 1958; Jones, 1972) values calculated on a general corpus.

Concretely, we denote the word set $\mathcal{S}_{x_i}$ as the keywords within sentence x_i. In the following, we generate partially masked sentence x_i^*, where only the keywords in the sentence x_i are masked, and $f(x_i^*)$ as the corresponding redundant embedding. Since the model is forced to see only the non-masked context words, $f(x_i^*)$ actually encode the information from the trivial words. Thus, the sentence-level redundant sentence embedding is defined as follows:

$$x_i^* = \text{PartialMask}(x_i, \text{keyword}); \quad v_i^* = f(x_i^*) \tag{1}$$

3.2.2 Corpus-level Redundancy

Given an unlabeled sentence set $\mathcal{X} = \{x_i\}_{i=1}^n$, we feed all the sentences to the encoder f, and take the average embedding $\hat{v}$ as its corpus-level redundant embedding, which can be formally defined as follows:

$$\hat{v} = \frac{\Sigma_{i=1}^n f(x_i)}{n} \tag{2}$$

where $\hat{v}$ is the corpus-level redundant embedding of x_i.

3.3 Embedding Refinement

As illustrates above, two redundant embeddings from two levels are obtained. Then the embedding refinement operation can be formalized via the conceptually simple and empirically powerful element-wise subtraction operation, which is defined as follows:

$$v' = f(x_i) - \lambda_1 \cdot v_i^* - \lambda_2 \cdot \hat{v} \tag{3}$$

where $f(x)$ corresponds to the original embedding of x_i, and v_i^* and $\hat{v}$ represent the redundant embedding at two levels, respectively.

Since the two redundant embeddings typically do not contribute completely equal to the embedding v, directly subtracting without adaptive parameters would cause that mitigating sentences' redundancy too much or too little for a specific dataset.

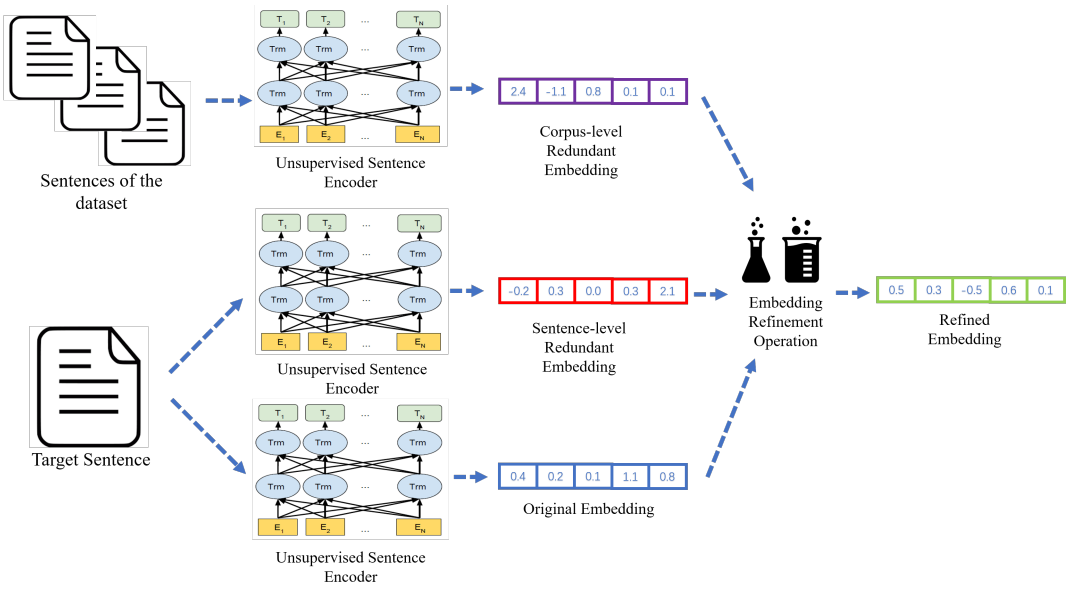


Figure 1: The overview of RepAL.

We introduce two independent hyper-parameters λ_1 and λ_2 to balance the two terms.

3.4 Theoretical analysis for corpus-level refinement

The operation that subtracts average embedding (i.e., $\hat{v}$) is widely used in deep neural networks (e.g., Batch Normalization), which controls the change of the layers' input distributions during training. In this section, we specifically delve into the effectiveness of such a subtraction operation in the sentence embedding improvement by analyzing the largest eigenvalue of the embedding matrix. We present two theorems to explain how corpus-level refinement diminishes the largest eigenvalue of the embedding matrix.

Let E be the embedding matrix for sentences $\{x_i\}_{i=1}^n$, and i-th row of E represents the sentence embedding of x_i. Assume we obtain the embedding matrix E through an unsupervised sentence encoder. Then the embedding matrix E^* after the corpus-level refinement operation is defined as follows:

$$E^* = E - \lambda \cdot \hat{E} \tag{4}$$

where $\hat{E}$ is a matrix whose each row is the same and assigned by average embedding $\hat{v} = \frac{1}{n}\sum_{i=1}^{n} v_i$. According to previous works (Wang et al., 2019b,a), the largest eigenvalue of E^*E^{*T} is dominant in controlling the quality of embedding space.

Therefore, controlling the largest eigenvalue of the matrix E^*E^{*T} can alleviate the degeneration problem and improves the quality of the learned sentence representations (Wang et al., 2019b).

Formally, given a real symmetric matrix $E^*E^{*T} \in R^{n\times n}$, the upper bound (Ostrowski, 1960; Zhan, 2005) of its largest eigenvalue σ is obtained as follows:

$$\sigma \leq \frac{1}{2}\left(nb + \sqrt{b^2 + (n^2-1)\,a^2}\right) \tag{5}$$

where a and b represent the max value and min value in the matrix E^*E^{*T}. Such a bound guides us in the direction of optimizing the sentence representation space and embedding performance. That is, if we can minimize the upper bound of the largest eigenvalue of the matrix by selecting a proper weight λ, then we can improve the quality sentence embeddings.

Next, we provide theoretical derivation that there exists suitable λ that can minimize the upper bound of the largest eigenvalue of E^*E^{*T}, thus improves the quality of sentence embeddings.

Theorem 1. *For a real matrix $E \in \mathcal{R}^{n\times d}$ and its row-average matrix $\hat{E}$. We denote W and W^* as EE^T and E^*E^{*T}, where E^* refers to Eq 4. Then $\exists\ \lambda$, s.t. upper_bound$(W^*, \rho) <$ upper_bound(W, ρ), where upper_bound(W, ρ) denotes the upper bound of the largest eigenvalue of the matrix W.*

The theorem above demonstrates the existence of λ and illustrates that if we choose such λ for the corpus-level refinement, then it is equivalent to

minimizing the largest eigenvalue of the embedding matrix.

Besides the theoretical analysis, we also launch numerical experiments to investigate whether such a conclusion still holds in downstream tasks, and conclude that we need to subtract the average embedding with adaptive weights, which are in Sec 5.2. The proofs of the theorems are deferred to the appendix A.

4 Experiments

In this section, we show that our method can be adaptive to various USE and improves their performance.

4.1 Baselines

To verify the effectiveness of our method, we evaluate RepAL on both Chinese and English settings. To investigate whether our method can be applied to various unsupervised sentence encoder (USE), we choose two kinds of encoders: vanilla USE and secondary trained USE. For vanilla USE, we select BERT (Devlin et al., 2019), RoBerTa (Liu et al., 2019), RoFormer (Su et al., 2021) and NEZHA (Wei et al., 2019) for Chinese; for English, we select BERT$_{base}$, BERT$_{large}$ (Devlin et al., 2019) and RoBERTa$_{base}$ (Reimers and Gurevych, 2019). Specifically, we name the secondary trained USE equipped with whitening (Huang et al., 2021), ConSERT (Yan et al., 2021), and SimCSE (Gao et al., 2021) as W-USE (e.g., W-BERT), C-USE (e.g., C-BERT), and Sim-USE (e.g., Sim-BERT), respectively. Results of Sim-USE and C-USE are from our implementation.

4.2 Benchmarks

- **Chinese**: We select five Chinese benchmarks[1] for evaluation. (1) AETC: A semantic similarity dataset related to customer service; (2) LCQMC: A dataset consisting problem matching across multiple domains; (3) BQ: a dataset consisting problem matching related to bank and finance; (4) PAWSX (Yang et al., 2019) : The dataset contains multilingual paraphrase and non-paraphrase pairs, we select the Chinese part; (5) STS-B: A Chinese benchmark labeled by semantic correlation between two sentences.

- **English**: We select STS task benchmarks as our English datasets. 7 datasets including STS 2012-2016 tasks (Agirre et al., 2012, 2013, 2014, 2015, 2016), the STS benchmark (Cer et al., 2017) and the SICK-Relatedness dataset (Marelli et al., 2014) are adopted as our benchmarks for evaluation.

[1] https://github.com/IceFlameWorm/NLP_Datasets

4.3 Training and Evaluation Settings

The vanilla USE in our experiments is the same as their original settings. Specifically, for Chinese USE, we select the [CLS] pooling; for English USE, we choose the average of outputs in the first and last layers. Besides, we keep the settings of whitening, ConSERT, and SimCSE the same as their original ones. As for hyper-parameters, we search the adaptive parameters on the validation set to select the weights of redundant embeddings, which can dynamically adapt to different datasets. The results are evaluated through weighted average Spearman correlation (Huang et al., 2021; Gao et al., 2021). Higher Spearman correlation indicates better capacity in sentence representation. In RepAL, we use the Jieba toolkit to extract the keywords within a sentence.

4.4 Performance on Chinese Benchmarks

As shown in Table 1, RepAL improves the baselines' performance in most cases. For example, RepAL produces 4.65%, 1.65%, 1.27%, and 0.88% improvement to BERT, W-BERT, C-BERT, and Sim-BERT, respectively. Generally, as USE becomes stronger, the improvements brought by RepAL decrease. Still, for strong baselines like C-BERT and Sim-BERT, RepAL still makes progress over them. Specifically, RepAL achieves 1.27% and 0.88% performance increase for C-BERT and Sim-BERT, indicating the effectiveness of RepAL on extremely strong baselines. Such experimental results demonstrate that RepAL is a general and powerful post-processing method for sentence embedding enhancement.

4.5 Performance on English Benchmarks

The experimental results on English benchmarks are listed in Table 2 and Table 3. As illustrated in Table 2, RepAL obtains improvements over the baselines averagely. Both results on Chinese and English benchmarks comprehensively demonstrate the effectiveness of RepAL and illustrate that RepAL is a plug-and-play method in unsupervised sentence representation learning.

Baseline	ATEC	BQ	LCQMC	PAWSX	STS-B	Avg
BERT	16.51→19.58	29.35→32.89	41.71→44.53	9.84→11.28	34.65→47.00	26.41→31.06(+4.65)
RoBERTa	24.61→27.00	40.54→39.51	70.55→70.98	16.23→16.98	63.55→64.01	43.10→43.70(+0.60)
RoFormer	24.29→25.07	41.91→42.56	64.87→65.33	20.15→20.13	56.65→57.23	41.57→42.06(+0.49)
NEZHA	17.39→18.98	29.63→30.53	40.60→41.85	14.90→15.43	35.84→36.68	27.67→28.69(+1.02)
W-BERT	20.61→23.29	25.76→29.83	48.91→50.01	16.82→16.96	61.19→61.46	34.66→36.31(+1.65)
W-RoBERTa	29.59→30.44	28.95→43.12	70.82→71.39	17.99→18.48	69.19→70.92	43.31→46.87(+2.56)
W-RoFormer	26.04→27.68	28.13→42.63	60.92→61.55	23.08→23.05	66.96→67.13	41.03→44.38(+3.35)
W-NEZHA	18.83→21.33	21.94→23.02	50.52→52.01	18.15→19.00	60.84→60.82	34.06→35.24(+1.18)
C-BERT	26.35→28.69	46.68→48.02	69.22→69.98	10.89→12.03	68.89→69.66	44.41→45.68(+1.27)
C-RoBERTa	27.39→28.43	47.20→47.14	67.34→67.98	09.36→10.55	72.02→71.80	44.66→45.18(+0.52)
C-RoFormer	26.24→27.68	47.13→47.63	66.92→67.85	11.08→11.65	69.84→69.73	44.24→44.91(+0.67)
C-NEZHA	26.02→26.73	47.44→48.02	70.02→70.63	11.46→11.80	68.97→69.53	44.78→45.34(+0.56)
Sim-BERT	33.14→33.48	50.67→51.14	69.99→72.44	12.95→13.58	69.04→69.55	47.16→48.04(+0.88)
Sim-RoBERTa	32.23→33.10	50.61→51.53	74.22→74.77	12.25→13.28	71.13→72.20	48.09→48.98(+0.89)
Sim-RoFormer	32.33→32.59	49.13→49.46	71.61→72.13	15.25→15.69	69.45→70.01	47.55→48.02(+0.47)
Sim-NEZHA	32.14→32.52	46.08→47.42	60.38→60.51	16.60→16.58	68.50→69.19	44.74→45.26(+0.52)

Table 1: The experimental results of RepAL on Chinese semantic similarity benchmarks. The numbers before → indicate the performance without RepAL and the numbers after → mean the performance with RepAL. Blue numbers indicate RepAL improves the baseline.

Baseline	STS-12	STS-13	STS-14	STS-15	STS-16	Avg
BERT	57.86→59.55	61.97→66.20	62.49→65.19	70.96→73.50	69.76→72.10	63.69→66.70(+3.01)
BERT$_l$	57.74→59.90	61.16→66.20	61.18→65.62	68.06→73.01	70.30→74.72	62.62→67.47(+4.85)
RoBERTa	58.52→60.88	56.21→62.20	60.12→64.10	69.12→71.41	63.69→69.94	60.59→65.41 (+4.82)
W-BERT	63.62→64.50	73.02→73.69	69.23→69.69	74.52→74.69	72.15→76.11	69.21→70.39 (+1.18)
W-BERT$_l$	63.62→63.90	73.02→73.41	69.23→70.01	74.52→75.18	72.15→75.89	69.21→70.39 (+1.18)
W-RoBERTa	68.18→68.85	62.21→63.03	67.13→67.69	67.63→68.23	74.78→75.44	67.17→68.43 (+1.26)
C-BERT	64.09→65.01	78.21→78.54	68.68→69.04	79.56→79.90	75.41→75.74	72.27→72.69 (+0.42)
C-BERT$_l$	70.23→70.70	82.13→82.54	73.60→74.12	81.72→82.01	77.01→77.58	76.03→76.48 (+0.45)
Sim-BERT	68.93→69.33	78.68→78.93	73.57→73.95	79.68→80.01	79.11→79.29	75.11→75.44 (+0.33)
Sim-BERT$_l$	69.25→69.60	78.96→79.30	73.64→73.92	80.06→80.31	79.08→79.42	75.31→75.61 (+0.30)

Table 2: The experimental results of RepAL on English semantic similarity benchmarks. 'Avg' indicates the average performance of all English benchmarks including STS-B and SICK-R in Table 3, and BERT$_l$ means BERT$_{large}$ during the experiments.

Baseline	STS-B	SICK-R
BERT	59.04→ 66.35	63.75→ 64.55
BERT$_l$	59.59→ 68.21	60.34→ 64.61
RoBERTa	55.16→ 65.75	61.33→ 63.61
W-BERT	71.34→ 71.45	60.60→ 62.61
W-BERT$_l$	71.34→ 69.56	60.60→ 65.00
W-RoBERTa	71.43→ 72.03	58.80→ 63.95
C-BERT	73.12→ 73.45	66.79→ 67.15
C-BERT$_l$	77.48→ 77.91	70.02→ 70.51
Sim-BERT	75.71→ 76.00	70.12→ 70.51
Sim-BERT$_l$	75.84→ 76.11	70.34→ 70.61

Table 3: The results of RepAL on STS-B and SICK-R

4.6 Hyper-parameters

Specifically, we search for λ_2 firstly and fix the best λ_2 to search for optimal λ_1 based on the dev set. Though there are two hyper-parameters in our method, searching for the optimal parameters is not computationally heavy since that our method is free of training. Generally. as the USE becomes stronger, the searched λ_1 and λ_2 become smaller.

5 Detailed Analysis and Discussion

The proposed RepAL enhances sentence embedding by filtering redundant information from two levels: sentence-level and corpus-level. Despite the presentations of the overall experiment results and analysis, the intrinsic properties of RepAL remain unclear. In this section, we illustrate the reasons why RepAL is effective in enhancing sentence embedding.

In Sec 5.1, we provide the evidence about the

Method	STS-12	STS-13	STS-14	STS-15	STS-16	STS-B	SICK-R	Avg
None	57.86	61.97	62.49	70.96	69.76	59.04	63.75	63.69
Tri-mask(individual)	58.12	63.05	63.71	71.02	71.19	61.29	63.90	64.61 (+0.92)
Key-mask(individual)	55.68	60.21	60.32	68.77	67.02	55.69	60.98	61.24 (-2.45)
Tri-mask(all)	59.55	66.20	65.19	73.50	72.10	66.35	64.55	66.70 (+3.01)
Key-mask(all)	57.76	62.56	63.35	70.49	70.93	61.31	63.50	64.27 (+0.58)

Table 4: The performance of two SR on the English benchmarks. 'Individual' indicates only applying SR and 'all' means combined with corpus-level refinement. The USE is BERT$_{base}$ here.

impact of trivial words in sentence embedding and show the capacity of our sentence-level embedding refinement. In Sec 5.2, we show why the corpus-level embedding refinement enhances sentence embedding and illustrate the relation between the largest eigenvalue and performance.

5.1 Sentence-level Refinement

There are two ablation studies for sentence-level refinement (SR): (1) we investigate the impact of trivial words w/o RepAL, which explains the necessity of removing such redundancy information and validates the effectiveness of SR. (2) We provide another sentence-level refinement solution by masking trivial words and then making comparisons, which further validates the effectiveness of SR.

5.1.1 Impact of trivial words

We set up experiments to investigate the capacity of sentence-level refinement individually and remove the corpus-level refinement. We first define the importance H of word $w \in x_i$ in semantic similarity calculation, which can be defined as follows:

$$H(x_i, x_i^-; w) = Sim(x_i, x_i^-) - Sim(x_i/w, x_i^-) \tag{6}$$

where x_i and x_i^- are a pair of sentences and x_i/w_i means deleting the word w_i from x_i. Note that we do not consider the words in x_i^- since it is equivalent to evaluation on more sentences. Then we define the set of trivial words within x_i as $S(x_i)$, which are unmasked by jieba. Thus we can define the **redundancy overlap ratio** $r(p_i)$ of a sentence pair $p_i = (x_i, x_i^-)$ as follows:

$$r(p_i) = \frac{|S(x_i) \cap T(x_i)|}{|T(x_i)|} \tag{7}$$

where $T(x_i)$ represents the top-5 words with highest importance H in x_i. $r(p_i)$ is a metric to reflect the impact of trivial words in semantic similarity between the sentence pair p_i, since higher $r(p_i)$ indicates more trivial words are important towards semantic similarity calculation. We randomly sample 300 sentence pairs from STS-B (Cer et al., 2017) and select BERT as the USE, and we calculate the average **redundancy overlap ratio** $\hat{r} = \frac{\sum_{i=1}^{N} r(p_i)}{N}$ w/o SR. The results show that $\hat{r}$ reaches 10.2% without SR, after applying SR, $\hat{r}$ drops to 7.1%[2] The results demonstrate that SR diminishes the impact of trivial words when measuring semantic similarity.

Moreover, we select some representative words and evaluate their importance w/o RepAL. As shown in Table 5, the results show that our SR indeed diminishes the impact of such trivial words when calculating semantic similarity.

Word	No Refinement	With Refinement	Δ
the	1.02	0.56	-0.46
a	0.98	0.43	-0.55
to	0.59	0.32	-0.27
in	0.68	0.21	-0.47
some	0.60	0.31	-0.29
with	0.72	0.24	-0.48
and	0.99	0.61	-0.38

Table 5: The importance of trivial words w/o sentence-level refinement. Δ means the importance change.

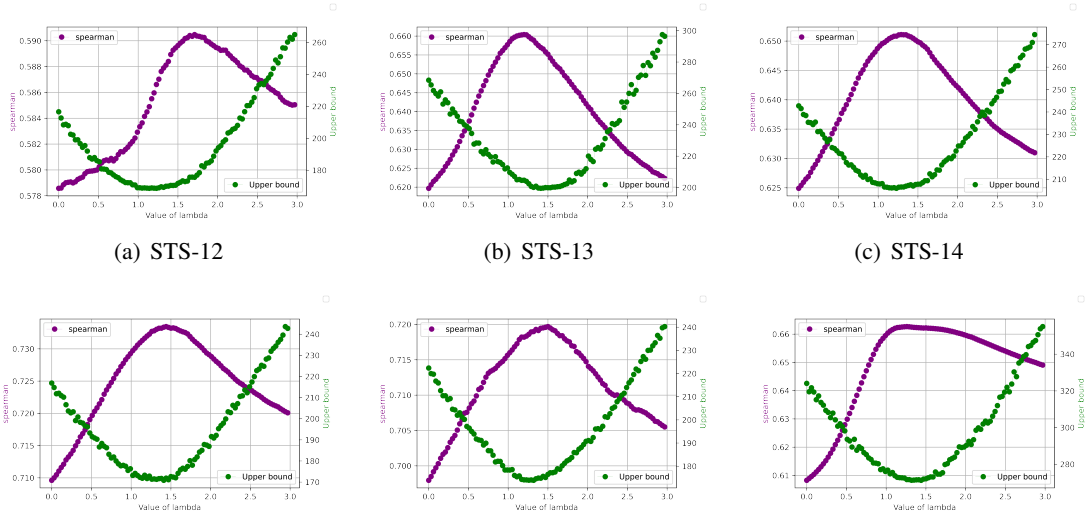
5.1.2 Comparison with Another Sentence-level Refinement Solution

Our SR generates redundant embedding by masking keywords and subtracts the redundant embedding from original embedding with an adaptive factor.

$$x_i^* = \text{PartialMask}(x_i, \text{keywords})$$
$$v_i = f(x_i) - \lambda f(x_i^*)$$

Then we propose another SR which masks the trivial words within the sentence and directly utilize

[2] $\hat{r}$ changes since the inputs during similarity calculation have changed when SR activates. After SR. Eq 6 becomes $H(x_i, x_i^-; w) = Sim(G(x_i), G(x_i^-)) - Sim(G(x_i/w), G(x_i^-))$ where $G(\cdot)$ represents SR operation.



(a) STS-12 (b) STS-13 (c) STS-14

(d) STS-15 (e) STS-16 (f) STS-B

Figure 2: Relation of upper bound of eigen value of embedding matrix and performance.

the embedding of sentence composed of remaining keywords, which can be defined as follows:

$$v_i = f(\text{PartialMask}(x_i, \text{trivial words})) \quad (8)$$

According to the words they mask, we name the two methods as tri-mask and key-mask.

We compare their performance on the English benchmarks. As shown in Table 4, tri-mask significantly outperforms key-mask. Specifically, when key-mask is applied individually, the performance even degrades worse than vanilla BERT. Intuitively, the key-mask performs poorly because it deletes all the trivial words within the sentence, which hurts the linguistic properties of the sentence (e.g., syntax information). In contrast, tri-mask eliminates the impact of trivial words through the subtraction with an adaptive weight factor, which preserves syntax information well.

5.2 Corpus-level Refinement

To investigate whether corpus-level refinement diminishes the upper bound of the eigenvalue of embedding E^*, we make numerical experiments to dive into the relationship between the performance (Spearman correlation), λ and the upper bound of the largest eigenvalue of E^*.

Specifically, we launch the experiments on six English benchmarks with BERT$_{base}$. As shown in Figure 2, when performance rises at peak, the upper bound of the largest eigenvalue of the embedding matrix E^* is around the minimum, showing a coincidence between the two. The numerical results show that the corpus-level refinement enhances sentence embedding since it diminishes the largest eigenvalue of E^*. Previous methods (Huang et al., 2021) is equivalent to subtracting the average vector with $\lambda = 1.0$, which fails to suppress the largest eigenvalue of embedding matrix extremely. However, our method chooses to subtract a larger λ with adaptive weight, further suppressing the upper bound of the largest eigenvalue of the embedding matrix. The results show that the average embedding subtraction needs an adaptive weight. Moreover, this also illustrates why our method can still improve the performance on W-BERT with substantial progress.

6 Conclusion

In this paper, we propose RepAL, a universal method for unsupervised sentence representation enhancement that combines with various USEs. Based on the idea that redundant information is contained in USE, RepAL extracts then refines redundant information for the sentence embedding at sentence-level and corpus-level. Sentence-level refinement aims at mitigating the impact of trivial words within the sentence; corpus-level refinement explicitly diminishes the upper bound of the largest eigenvalue of the embedding matrix. Combining them into one, RepAL successfully achieves improvements on both Chinese and English benchmarks and is proved to be a simple and plug-and-play method in modern techniques for unsupervised sentence representation.

References

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Inigo Lopez-Gazpio, Montse Maritxalar, Rada Mihalcea, et al. 2015. Semeval-2015 task 2: Semantic textual similarity, english, spanish and pilot on interpretability. In *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)*, 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 8th international workshop on semantic evaluation (SemEval 2014)*, pages 81–91.

Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez Agirre, Rada Mihalcea, German Rigau Claramunt, and Janyce Wiebe. 2016. Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In *SemEval-2016. 10th International Workshop on Semantic Evaluation; 2016 Jun 16-17; San Diego, CA. Stroudsburg (PA): ACL; 2016. p. 497-511.* ACL (Association for Computational Linguistics).

Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. 2012. Semeval-2012 task 6: A pilot on semantic textual similarity. In ** SEM 2012: The First Joint Conference on Lexical and Computational Semantics–Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 385–393.

Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2013. * sem 2013 shared task: Semantic textual similarity. In *Second joint conference on lexical and computational semantics (* SEM), volume 1: proceedings of the Main conference and the shared task: semantic textual similarity*, pages 32–43.

Daniel Cer, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia. 2017. Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation. *arXiv preprint arXiv:1708.00055*.

Seungtaek Choi, Haeju Park, Jinyoung Yeo, and Seungwon Hwang. 2020. Less is more: Attention supervision with counterfactuals for text classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6695–6704.

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

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Felix Hill, Kyunghyun Cho, and Anna Korhonen. 2016. Learning distributed representations of sentences from unlabelled data. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1367–1377.

Junjie Huang, Duyu Tang, Wanjun Zhong, Shuai Lu, Linjun Shou, Ming Gong, Daxin Jiang, and Nan Duan. 2021. Whiteningbert: An easy unsupervised sentence embedding approach. *arXiv preprint arXiv:2104.01767*.

Karen Sparck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*.

Ryan Kiros, Yukun Zhu, Russ R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *Advances in neural information processing systems*, pages 3294–3302.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196. PMLR.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Lajanugen Logeswaran and Honglak Lee. 2018. An efficient framework for learning sentence representations. In *International Conference on Learning Representations*.

Hans Peter Luhn. 1958. The automatic creation of literature abstracts. *IBM Journal of research and development*, 2(2):159–165.

Rabeeh Karimi Mahabadi, Yonatan Belinkov, and James Henderson. 2020. End-to-end bias mitigation by modelling biases in corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8706–8716.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, Roberto Zamparelli, et al. 2014. A sick cure for the evaluation of compositional distributional semantic models. In *Lrec*, pages 216–223. Reykjavik.

Yu Meng, Chenyan Xiong, Payal Bajaj, Saurabh Tiwary, Paul Bennett, Jiawei Han, and Xia Song. 2021. Coco-lm: Correcting and contrasting text sequences for language model pretraining.

Alexander Markowich Ostrowski. 1960. On the eigenvector belonging to the maximal root of a nonnegative matrix. *Proceedings of the Edinburgh Mathematical Society*, 12(2):107–112.

Matteo Pagliardini, Prakhar Gupta, and Martin Jaggi. 2018. Unsupervised learning of sentence embeddings using compositional n-gram features. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 528–540.

Chen Qian, Fuli Feng, Lijie Wen, Chunping Ma, and Pengjun Xie. 2021. Counterfactual inference for text classification debiasing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5434–5445.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992.

Richard Socher, Eric Huang, Jeffrey Pennin, Christopher D Manning, and Andrew Ng. 2011. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. *Advances in neural information processing systems*, 24.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Jianlin Su, Yu Lu, Shengfeng Pan, Bo Wen, and Yunfeng Liu. 2021. Roformer: Enhanced transformer with rotary position embedding. *arXiv preprint arXiv:2104.09864*.

Dilin Wang, Chengyue Gong, and Qiang Liu. 2019a. Improving neural language modeling via adversarial training. In *International Conference on Machine Learning*, pages 6555–6565. PMLR.

Dong Wang, Ning Ding, Piji Li, and Haitao Zheng. 2021. Cline: Contrastive learning with semantic negative examples for natural language understanding. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2332–2342.

Lingxiao Wang, Jing Huang, Kevin Huang, Ziniu Hu, Guangtao Wang, and Quanquan Gu. 2019b. Improving neural language generation with spectrum control. In *International Conference on Learning Representations*.

Junqiu Wei, Xiaozhe Ren, Xiaoguang Li, Wenyong Huang, Yi Liao, Yasheng Wang, Jiashu Lin, Xin Jiang, Xiao Chen, and Qun Liu. 2019. Nezha: Neural contextualized representation for chinese language understanding. *arXiv preprint arXiv:1909.00204*.

Yuanmeng Yan, Rumei Li, Sirui Wang, Fuzheng Zhang, Wei Wu, and Weiran Xu. 2021. Consert: A contrastive framework for self-supervised sentence representation transfer. *arXiv preprint arXiv:2105.11741*.

Yinfei Yang, Yuan Zhang, Chris Tar, and Jason Baldridge. 2019. Paws-x: A cross-lingual adversarial dataset for paraphrase identification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3687–3692.

Xingzhi Zhan. 2005. Extremal eigenvalues of real symmetric matrices with entries in an interval. *SIAM journal on matrix analysis and applications*, 27(3):851–860.

Yan Zhang, Ruidan He, Zuozhu Liu, Kwan Hui Lim, and Lidong Bing. 2020. An unsupervised sentence embedding method by mutual information maximization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1601–1610.

Xiang Zhou and Mohit Bansal. 2020. Towards robustifying nli models against lexical dataset biases. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8759–8771.

A Proof of Theorem 1

Formally, we denote σ_W and σ_{W^*} for $\rho(W)$ and $\rho(W^*)$ for brevity. Then we have:

$$W = \left\| E^T E \right\| \tag{9}$$

$$W^* = \left\| (E - \lambda \hat{E})^T (E - \lambda \hat{E}) \right\| \tag{10}$$

Based on the properties of singular value, given a matrix A, for an arbitary vector x, we have:

$$\|Ax\| \leq \sigma_A \|x\| \tag{11}$$

where σ_A represent the largest singular value of A. Therefore, for a unit eigenvector v of W^*, we have the following derivation:

$$\begin{aligned}
&\sigma_{W^*} = \|\sigma_{W^*} v\| = \left\| (E - \lambda \hat{E})^T (E - \lambda \hat{E}) v \right\| \\
&\leq \left\| E \hat{E}^T v \right\| - \left\| \lambda E \hat{E}^T v \right\| \\
&- \left\| \lambda \hat{E} E^T v \right\| + \left\| \lambda^2 \hat{E} \hat{E}^T v \right\| \\
&\leq \sigma_E^2 - 2\lambda \sigma_E \sigma_{\hat{E}} + \lambda^2 \sigma_{\hat{E}}^2 \\
&= (\sigma_E - \lambda \sigma_{\hat{E}})^2
\end{aligned} \tag{12}$$

where σ_E and $\sigma_{\hat{E}}$ represent the largest singular value of E and $\hat{E}$, respectively. Similarily, we can obtain the upper bound for σ_{W^*}, which is σ_E^2. Let λ be $\frac{c}{\sigma_{\hat{E}}}$ where c is a positive constant, the upper bound of σ_{W^*} is apparently lower than σ_W, which completes the proof.