International Journal of Machine Learning and Cybernetics (2023) 14:2591–2605
https://doi.org/10.1007/s13042-023-01784-y

ORIGINAL ARTICLE

Data augmentation using Heuristic Masked Language Modeling

Xiaorong Liu[1] · Yuan Zhong[1] · Jie Wang[2] · Ping Li[1]

Received: 29 July 2022 / Accepted: 18 January 2023 / Published online: 30 January 2023

Abstract
Data augmentation has played an important role in generalization capability and performance improvement for data-driven deep learning models in recent years. However, most of the existing data augmentation methods in NLP suffer from high manpower consumption or low promotion, which limits the practical applications. To this end, we propose a simple yet effective approach named Heuristic Masked Language Modeling(HMLM) to obtain high-quality data by introducing mask language modeling embedded in pre-trained models. More specifically, the HMLM method first identifies the core words of the sentence and masks some non-core fragments in the sentence. Then, these masked fragments will be filled with words created by the pre-trained model to match the contextual semantics. Compared with the previous data augmentation approaches, the proposed method can create more grammatical and contextual augmented data without a heavy cost. We conducted experiments on typical text classification tasks e.g., intent recognition, news classification and sentiment analysis separately. Experimental results demonstrate that our proposed method is comparable to state-of-the-art data augmentation approaches.

Keywords Data augmentation · Mask language modeling · Pre-trained models · Text classification

1 Introduction

The superior performance of the deep neural network model is known for having heavy data dependence, which might not be available in resource-lean scenarios [1]. Data augmentation (DA) is a widely used technique to increase the size of the training data [2–4]. In data augmentation, new labeled instances are created by modifying the features of existing instances with transformations that do not change the label of an instance. Data augmentation methods for computer vision include image translation, flipping, scaling, separation, etc. However, these methods cannot be transferred to natural language processing tasks directly. The difference between the two can be attributed to the discreteness of text data. For the text data, every word plays an important role in expressing the whole meaning, and the order of words is also consequential to the semantics of sentences. Hence, inappropriate operations of augmentation on the text make changes to the semantic information, resulting in the failure to maintain the consistency of the label [5].

To address this issue, current text augmentation approaches such as back-translation [6, 7], word replacement [8], text augmentation based on contextual information [9, 10], and language generation method [11–13] have been attempted. The quality of the translation decoder determines the data generated by the back translation technique. The current issue is that not only are the majority of the translation decoding results inaccurate but also the usage of conversion models necessitates the utilization of extra computer resources. In terms of replacement methods, hey are primarily based on random substitution and dropping words. Although this kind of method has a minimal time cost, it may lose the semantic structure and order of the original training data, which brings about changes in the augmented semantic labels. Contextual information-based data augmentation is also called conditional data augmentation. This kind of method is based on the pre-trained language model and

✉ Yuan Zhong
strangeryy0202@163.com

Xiaorong Liu
Hsiaorliu@gmail.com

Jie Wang
wangjie_self@163.com

Ping Li
dping.li@gmail.com

[1] School of Computer Science, Southwest Petroleum University, Chengdu, China

[2] School of Computing and Artificial Intelligence, Southwest Jiaotong University, Chengdu, China

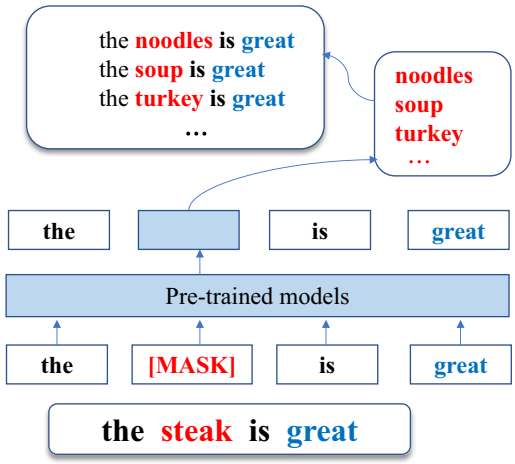

Fig. 1 HMLM data augmentation, when a sentence "the steak is great" is augmented by replacing only steak with words predicted based on the context, and the core word "great" is kept unchanged

integrates the label information into the neural network for fine-tuning. In addition, language generation models are also used for data augmentation. However, when the language generation model performs text augmentation, it generates text without restriction and may not be good at saving label information.

As discussed, keeping the semantic structure and labels correct makes text augmentation more challenging. The text label is a word that highly generalizes the text content. Each token in the text sequence has different contributions to the generation of the label. We can keep the label correct if the word with the greatest contribution is reserved when conducting data augmentation operations. Therefore, this paper is motivated by the desire for the model to create augmented data with reserved labels at relatively low cost. As a result, we propose a simple and effective data augmentation method, Heuristic Masking Language Modeling (HMLM), the execution example is shown in Fig. 1.

In particular, we first conduct core word recognition(CWR) for the text. Because of the different targets of the tasks, the core word may be an intention, emotional attitude word, or a term necessary for classification. Following that, one of these non-core words in the text is randomly masked. And then, the pre-trained language model is allowed to make predictions to achieve the purpose of text augmentation. In a nutshell, our contribution can be summarized as follows:

(1) We propose an effective and low-cost data augmentation approach to generate rich training data, which is according to the prior knowledge and contextual information to mask, and fill the masked words in the sentence heuristically through utilizing mask language modeling based on pre-trained language models.
(2) We adopted different ways to conduct core word recognition (CWR) according to the characteristics of specific tasks, which guarantee the recognition accuracy of the core word, and make the created data keep the same distribution as the original data.
(3) Experiments show that the method outperforms baseline methods in different natural language processing applications. In particular, it is a robust improvement on the problem of data label changes in the process of data augmentation compared to other methods. The rest of the paper is organized as follows. Section 2 introduces related work in the data augmentation methods. Section 3 introduces the method that we proposed. Section 4 describes the experiment setup. Section 5 introduces comparative experiment and discussion. Section 6 introduces the summary and future works.

2 Related work

2.1 Data augmentation

Deep learning has performed remarkably well on many natural language processing tasks recently. The superior performance of deep neural network models has heavy data dependence so a large amount of data is needed to minimize overfitting. However, the size and quality of labeled data for many real-world NLP (Natural Language Processing) applications are frequently constrained. Hence, data augmentation has become an effective way to augment the size and quality of the training data; even it is crucial to the successful application of deep learning models on data-driven projects. In addition to some existing traditional data augmentation techniques, with the emergence of a considerable amount of the work of pre-trained models, researchers are also attempting to employ pre-trained models to bring a new perspective to data augmentation.

Word replacement The principle of word replacement is to create training data similar to the original data by injecting noise into the original data. Jason Wei and Kai Zou [14] proposed the EDA method to replace words randomly by thesaurus. Although EDA has made some achievements, it still has a few shortcomings that cannot be ignored. On the one hand, synonyms need to be defined manually, which takes a significant amount of time. On the other hand, the kind of selective substitution without emphasis may give rise to the loss of the semantic structure of the augmented data. UDA optimizes EDA's random word processing strategy, which uses the probability of a negative correlation with TF-IDF to extract words in each text to determine whether to replace [1]. Dai et al. [15] proposed for sentence-level and sentence-pair NLP tasks, using binomial distribution to

determine whether each token is replaced. The replacement method is to randomly select another token with the same label. Word replacement method also chooses words that share the same morphology [16, 17].

Other generic word replacement methods include word dropout, which randomly drops out any number of a word's occurrences, and not just none or all words [18, 19]. Reward Augmented Maximum Likelihood (RAML) essentially replaces certain words in the target sentence with other words in the target vocabulary [20]. To the best of our knowledge, the performance of this method is relatively inferior to EDA in some respects.

Back-translation Back translation signifies that the original data is translated into other languages, and then back into the original language, the target language. If the intermediate languages are different, the final augmented data may be more diverse. Back translation technology was primarily employed by early researchers to increase the performance of the neural network translation model (NMT). Rico Sennrich et al. [21] proposed a general data augmentation method in machine translation (MT) that allows the system to incorporate monolingual training data. For instance, when training an English→French system and the monolingual French text is translated to English using a French → English system, the synthetic parallel data can be used for training subsequently. Back-translation can also be used for paraphrasing [22], and it has been used for data augmentation for QA [13]. Adams Wei Yu et al. [23]used the back-translation technology as a special data augmentation technology to optimize the performance of the question and answer model.

Contextual information-based This work was first proposed by Kobayashi and Sosuke [9], and they used a contextual information augmentation method based on a bidirectional language model. A contextualized language model usually captures the information of words and manages the morpho-syntactic variations typical of handwritten notes. In order to ensure the consistency of the label after the augmentation, the researchers embedded the label information in the LM hidden layer. Similar to Contextual augmentation, CBERT replaced the bidirectional LM with BERT and also fine-tuned the BERT, introducing the label information of the original text to make sure that augmented samples have the same label attributes as the original samples [10].

Language generation model LAMBADA used a method of synthesizing labeled data [2]. It was first pre-trained on a huge volume of the text so that the model can capture the structure of the language, which could produce coherent sentences. They fine-tuned the model on a small number of datasets for different tasks, and the new sentences were then generated using the fine-tuned model. Finally, training the classifier on the same small dataset and filtering it to ensure that the existing small data and the newly generated data have a similar distribution.

Other methods In addition to the popular approaches mentioned above, there are also alternative methods that contribute to text data augmentation. Yitong Li et al. [24] proposed a linguistically motivated method to text application customization, which is based on injecting noise into the input text. Michihiro et al. [25] proposed to generate samples by increasing the reverse disturbance at the input-end in the direction of significantly increasing the classifier's loss function. Moustafa Alzantot et al. [26] utilized an unrestricted end-to-end solution to efficiently generate adversarial texts.

2.2 Pre-trained models for NLP

Recently, substantial work has shown that pre-trained models (PTMs) on the large corpus can learn universal language representations, which are beneficial for downstream NLP tasks and can avoid training a new model from scratch [27]. With the help of the representation extracted from the PTM in the large unannotated corpus, there was a significant performance improvement on many NLP tasks. Among them, self-encoding language models, like the BERT series, are developed by the transformer, which proves the importance of the attention mechanism. It discarded the traditional convolutional neural networks(CNNs) and recurrent neural networks(RNNs) in the encoder-decoder and consists of only attention mechanisms and Feed Forward Neural Networks.

BERT. It used a bidirectional transformer block connection and introduced Masked Language Modeling(MLM) pre-trained targets to enable it to obtain context-related bidirectional feature representations [28], introducing Next Sentence Prediction (NSP) pre-trained targets to make the model good at handling sentence or paragraph matching task.

ERNIE 2.0 This model introduced multi-task learning(interacting with a priori knowledge base) pre-trained so that the model could learn more language knowledge from different tasks [29]. The main method was to build an incremental learning model, which continuously updated the pre-trained model through multi-task learning. This continuous alternating learning paradigm will not make the model forget the previously learned language knowledge.

Chinese-Bert-wwm It utilized the whole word masking strategy for Chinese BERT, which improved the problem that BERT's Word Mask mechanism may affect the meaning of words in Chinese [30].

RoBERTa It made some adjustments based on BERT[1]. It removed the next predicted loss, dynamically adjusted the mask mechanism, and the training sequence was longer. In addition, it took longer to train and needed more training data. Such a change may introduce more prior knowledge for data augmentation.

Different from the previous data augmentation work using the pre-trained model and fine-tuning pattern. The proposed HMLM adopts the LM_mask function of the self-encoding language model to mask the non-core words randomly in text and uses the pre-trained model to predict the mask part of the original data directly to obtain the text, which maintains the consistent label with the original.

3 Heuristic Masked Language Modeling

3.1 Problem definition

Text classification refers to the process of assigning predefined labels to text and is an instance of the supervised learning problem for text data. By this definition, a training dataset is given:

$D_{train}=\{(x_i, y_i)\}_{i=1}^{n}$ containing n labeled sentences. Here $p(X,Y)$ indecate the distribution over all training data pairs (x_i, y_i), and use a hat $\hat{p}(\hat{X}, \hat{Y})$ to denote distribution of augmented data. If $f(\cdot) \rightarrow \hat{p}(\hat{X}, \hat{Y}) \simeq p(X,Y)$, which means this data augmentation fuction $f(\cdot)$4ss is effective. There are differents ways to measure the quality of a data augmentation function, but the most straightforward way is to measure whether it improves the accuracy of the downstream tasks. If the augmented data can significantly improve the performance of the downstream tasks classifier, it can be considered that data augmentation improves the model robustness, making the model pay more attention to the semantic information of the text and to the local noise of the text no longer sensitive. In the text classification task, we use the cross-entropy [31] loss function to update the parameters. Its formula can be expressed as:

$$J_\theta = -\sum_{i=1}^{N} p(x_i) \log q(x_i) \tag{1}$$

where $p(x_i)$ denotes the true label distribution, and $q(x_i)$ denotes the predicted label distribution.

3.2 Heuristic mask strategy

To the best of our knowledge, the concept of the heuristic was first proposed in the optimization algorithm. It means that in a random optimization process, individuals can use their own or global experience to formulate their own search strategies. In a nutshell, the emphasis of the heuristic is how to better use global or self-information. Similarly, we expect that the data augmentation model will be able to use global information with label information to maintain the correct semantic tags instead of embedding tags into the input for retraining.

In some conventional NLP tasks, their labels are determined by the core words, such as the sentiment analysis task, it is the emotional attitude words that determine their labels, so we regard the emotional attitude words as core words for sentiment analysis. When performing the mask decision, it is clear that improper mask positions will affect the model's use of global information. The random mask strategy may mask the core word so that the pre-trained model may replace these significant words when filling the mask part, resulting in label variation. On the contrary, if core words are retained, the pre-trained model can make good use of the global information with core words, thus keeping the label invariant. In order to maintain the label correctness of the augmented data, the proposed approach conducts core word recognition to bypass core words before using the pre-trained model for mask-prediction. This strategy utilize the prediction function of the pre-trained model in the data augmentation stage, which does not require additional training at all, which means our approach is low-cost.

The procedure of the proposed approach is shown in Fig. 2. There are three subgraphs from bottom to top, which respectively illustrate the specific operation of each stage of the HMLM algorithm. Subgraph (a) shows the stage of CWR for the input sentence. The core word "news" is picked as recognition results. Then, the subgraph (b) shows the mask operation; some non-core words are arbitrarily masked after we bypass the core word. Next, the subgraph (c) shows that the pre-trained model is used to predict some words to replace these masked words to make sure the context of the sentence is clear and coherent in the meantime. As shown in subgraph (b), due to the different positions of the mask words, more and different augmented data can be obtained by the predicting stage.

3.3 Core word recognition

The significance of ensuring the correctness of semantic tags for data augmentation has already been discussed. In common NLP tasks, semantic tags are usually determined by several core words. For example, the labels of intent recognition are determined by intent keywords. In news classification, it is determined by keywords, and in sentiment analysis, some emotional words determine the emotional direction. Ignoring core words and using random replacement words for data augmentation, such as EDA, may cause

[1] https://github.com/brightmart/roberta_zh.

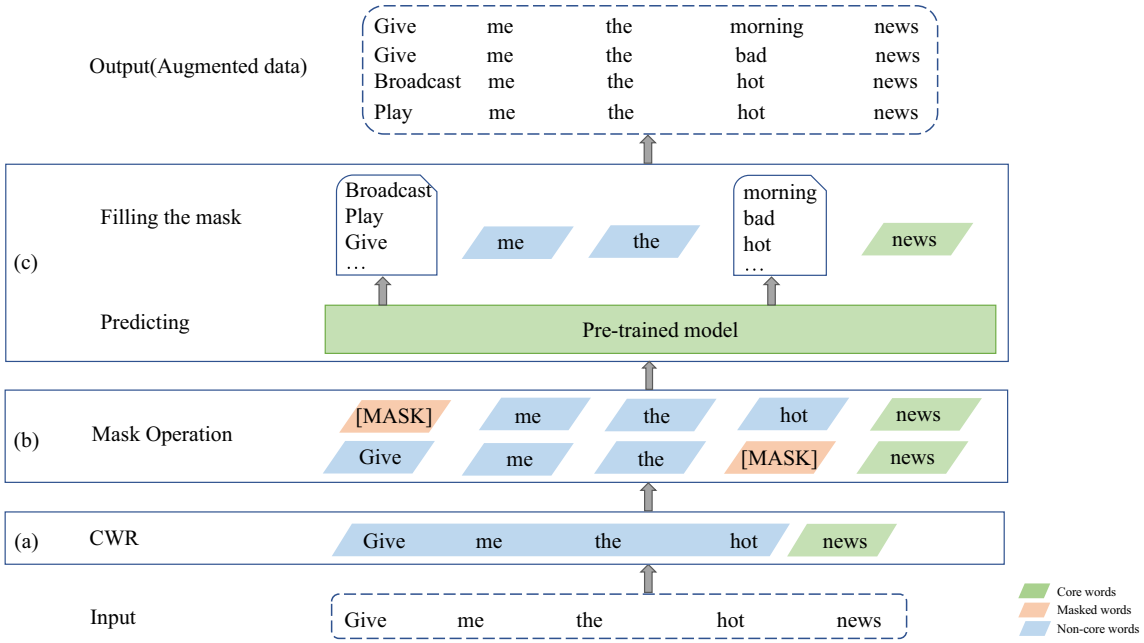

Fig. 2 The procedure of data augmentation using HMLM. For instance, the label of "Give me the hot news" is "News." If the word "news" is replaced, the label changes accordingly. As a result, the word "news" is regarded as the core word, and the words in the sentence other than "news" can be regarded as non-core words. We recommend data augmentation for any words that do not change the label to ensure label continuity

the expanded data to obtain wrong labels. In this paper, as shown in Fig. 2a, we first recognize the core words in the input text. In particular, the different ways are applied to distinguish the core words according to the characteristics of specific tasks, such as textrank, BiLSTM+Attention, etc. Non-core words of training data are masked to construct samples when using the pre-trained model, as shown in Fig. 2b.

3.4 Predicting using mask language modeling

The mask language model is not a rigorous language model, but a way to train a language model. It is an example of autoencoding language modeling that reconstructs the output from a corrupted input. We typically mask one or more words in a sentence and have the model predict those masked words given the context of the sentence. By training the model with such an objective, it can essentially learn certainly, but not all, statistical properties of word sequences. Mask language model (MLM) was first proposed by Tylor as a cloze task in literature [32], with the idea that the better readability of an article, the lesser people made mistakes in guessing the hidden word. Devlin et al. [28]adapted this task as a novel pre-trained task to overcome the drawback of the standard unidirectional language model. The main way is to supervise themselves, for example, removing several words in a paragraph and using their context to predict the masked words. Similar to BERT, it randomly replaces 15% of words of the sentence with mask tags and, after that, predicts the masked words. Generally speaking, modeling the probability of natural language generation is the goal of the preview model. In terms of the bidirectional language model, given a sequence of n words, $X=\{x_1,x_2,...,x_n\}$, the probability that the forward language model [27] predicts the sequence is:

$$p(x_1, x_2, \ldots, x_n) = \prod_{i=1}^{n} p(x_i \mid x_1, x_2, \ldots, x_{i-1}) \tag{2}$$

The backward language model predicts the probability of a sentence as:

$$p(x_1, x_2, \ldots, x_n) = \prod_{i=1}^{n} p(x_i \mid x_{i+1}, x_{i+2}, \ldots, x_n) \tag{3}$$

In the training stage, the loss function is to allow the language model to attempt to restore its original input. Different from the training phase goal, we do not have to reconstruct the original input; on the contrary, we hope that the model's prediction and the original input have a certain difference, and this difference enriches our input sentence, the flow chart of prediction phase is shown in Fig. 2c.

Some augmented examples are given by experimenting with our data augmentation algorithm on three tasks: (1) intent detection; (2)news classification; (3)sentiment analysis. For each task, perform the following steps: (1) core word recognition; (2) using the mask language modeling

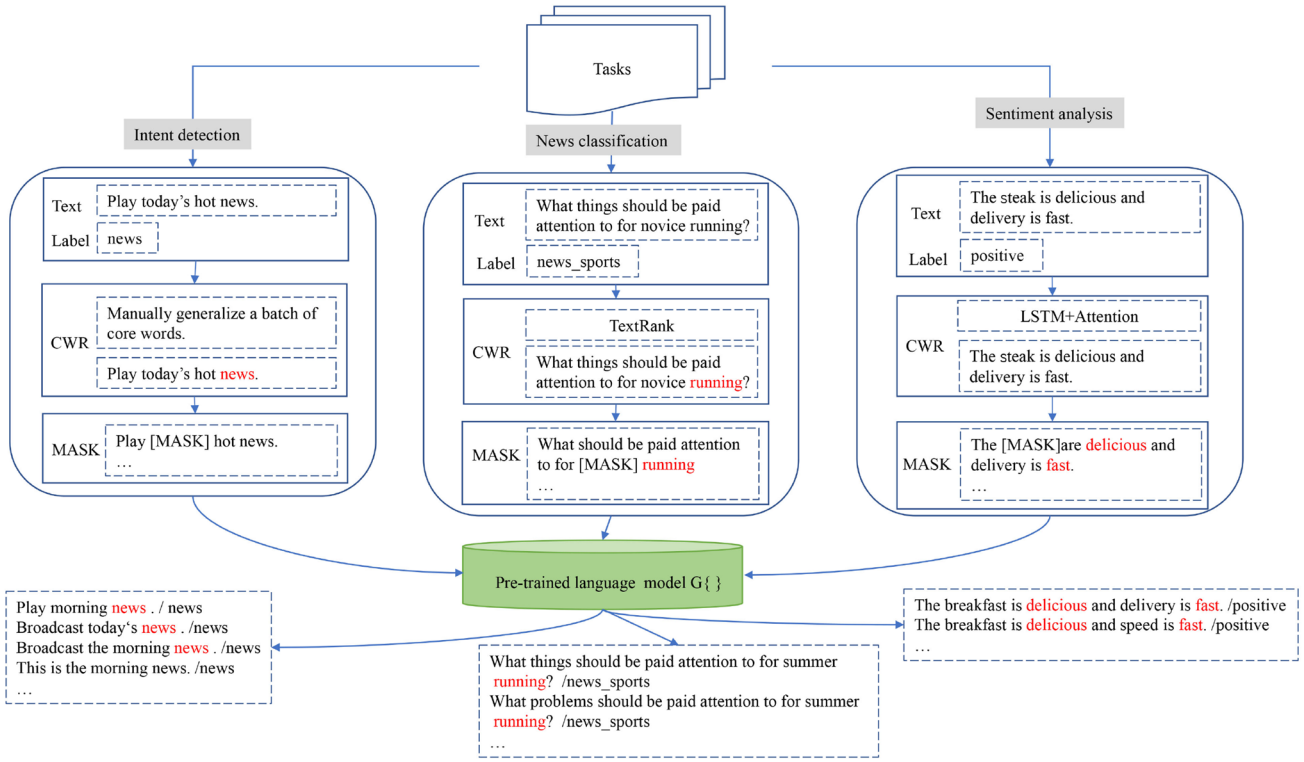

Fig. 3 Examples of data augmentation process diagram where CWR means core word recognition

for data augmentation; (3) verifying that the augmented data improves the performance of the classifier. As shown in Fig. 3, we can see that in the core word recognition stage, we utilize different approaches to identify core words for these three tasks.

We mark all the recognized core words in red to visually observe the changes in the data. After that, we can see that the red words are avoided mask when the mask token is added. For each task, the examples given in the figure are derived from different mask positions under pre-trained language models G{$BERT-base$}. To obtain more diverse augmented data, with the mask position unchanged, only the pre-trained model needs to be changed. In addition, during the execution of mask-prediction, the original input may also be predicted. Such samples will be regarded as invalid augmented data.

Simultaneously, the pseudo-code demonstrates the process of data augmentation, as shown in Algorithm 1. It is necessary to interpret some of the notations in Algorithm 1. Given a training dataset D_{train}, contain labeled sentences to generate augmented dataset D_{aug}. For line 3, the variable **s** represents each sentence among in D_{train}. For line 5, l_s denotes the sentence length of D_{train}, and l_c indicates the length of the core word in each sentence. For line 11, $\hat{s}$ represents the sentence generated by the model.

Algorithm 1 Data augmentation algorithm

Require: Training dataset D_{train},
1: G$\in$ {$BERT-base$, $ERNIE2.0$, $BERT-wwm$, $RoBERTa$},
2: Core word recognition algorithm Λ;
Ensure: Augmented dataset D_{aug};
3: **for** each **s** in D_{train} **do**
4: extract core words using Λ
5: k $\in \{l_s - l_c\}$
6: **for** each model in G{} **do**
7: mask k non-core words arbitrarily
8: do generate sentences $\hat{s}$
9: **end for**
10: **end for**
11: **while** $\hat{s}$!= **s** **do**
12: add $\hat{s}$ to D_{aug}
13: **end while**
14: return D_{aug}

4 Experiment

4.1 Tasks and datasets

Three type of text tasks are designed in experiments: intent recognition, news classification, and sentiment analysis.

Table 1 Datasets details

Task	Dataset	Classes	Avg number of words
Intent detection	intent_news	2	8
News classification	toutiao_news[2]	15	22
Sentiment analysis	Sentiment[3]	2	15

[2]https://github.com/BenDerPan/toutiao-text-classfication-dataset

[3]https://github.com/SophonPlus/ChineseNlpCorpus/blob/master/datasets/waimai_10k

Three Chinese single-label multi-classification datasets description is shown in Table 1. The dataset for intent detection comes from the online data collected by the company's dialogue system, we named it intent_news, which contains 600 news-type intent and 2000 non-news-type intent. We select toutiao_news of the public dataset for news classification, and judged whether the news is one of 15 categories such as news_fiance, news_culture, etc, a total of 73360 samples. It can be classified as a long text classification task. The sentiment dataset comes from user reviews collected by a takeaway platform, of which 4000 are positive and about 8000 are negative. we randomly take 80% of samples as the training set and the last 20% as the testing set.

4.2 Models and experimental setup

Baseline We consider five models as our baseline.

(1) EDA [14] model is a random word replacement method based on the thesaurus, which has proved to be effective in augmenting scenarios with small data sets.
(2) HDA [33] implemented a hierarchical data augmentation strategy by augmenting texts at word-level and sentence level respectively and utilizes attention mechanism to distill (crop) important contents from texts hierarchically as summaries of texts.
(3) UDA [1] model introduced TF-IDF to measure the importance of a word to a sentence. In essence, it can be regarded as introducing strong prior knowledge on the basis of EDA and then replacing synonyms according to determined keywords.
(4) CBERT [10] model retrofitted BERT to conditional BERT by introducing extra label-conditional constraint to the mask language model.
(5) AugSBERT [34] utilized the cross-encoder to label a larger set of input pairs to augment the training data for the bi-encoder.

Parameter settings We present the parameter settings. The model and experiment settings include two aspects: classification model parameters and augmentation model parameters. On the one hand, the augmentation model parameters include the proportion of selected data β, the augmentation multiples γ for a single sentence, and the number of core words τ in each sentence. Among them, we stipulate that the ratio $\beta = \{10\%, 30\%, 50\%, 100\%\}$ of the three datasets are selected, and the augmentation multiples $\gamma = \{1, 2, 3\}$ is achieved by masking different positions of the non-core word parts; and the number of core word τ in each sentence we set $\tau = \{1, 2\}$. Furthermore, the ratio of the mask in each sentence varies according to the length of the sentence. In general, only two consecutive characters are masked in a sentence at a time. In the predicting-filling stage, we also selected different pre-trained models to compare the filling effects, including BERT-base, ERNIE 2.0, BERT-wmm and RoBERTa. Finally, in order to explore the impact of data augmentation on the classifier, we feed the augmented data together with the original data into the classifier. On the other hand, for the three tasks, detailed parameters corresponding to the classification model are described below.

Intent detection Since intent recognition sentences are typically short, each sentence can reflect a strong intent. Hence, we generalized a batch of words related to intent according to the intent labels and took them as the core words for frecognition. In order to correspond to the realization of subsequent augmentation multiples, the number of core words is set to τ=1. For the TextCNN [35] model, we set lr=0.0001, batch_size=128, dropout=0.5, epochs=10, max_length=16, kernel=100, filter_size=(2,3,4), embedding_size[2]=200.

News classification Due to the long length of the news data, there may be multiple core words. Therefore, it is different from the intention detection, we have set τ =2. In news classification, the core words may determined by keywords, thus we use TextRank [36] to recognize core words. In terms of the TextCNN model, we set filter_size=(3,4,5), kernel=256, lr=0.001, batch_size=128, epochs=20, dropout=0.5, max_length=32, embedding_size[1]=200.

Sentiment analysis For sentiment analysis, emotional attitude words are used as the criteria for selecting core words. Owing comment may contain different aspects, and the aspects may have opposite polarities. We expect to extract the emotional attitude words contained in each aspect so as to avoid the emotional attitude words in the subsequent masking. The number of the core word is set to τ=1. For core word recognition, we employ the BiLSTM to represent

[2] https://ai.tencent.com/ailab/nlp/en/embedding.html.

Table 2 Comparison of accuracy improvement of intent detection

Multiple	Model		Training data proportion			
			β=10%	β=30%	β=50%	β=100%
γ=0	Original		52.17	74.00	84.33	96.96
γ=1	Baseline	EDA	59.55(+7.38)	77.77(+3.77)	86.79(+2.46)	96.67(-0.29)
		HDA	58.32(+6.15)	76.56(+2.56)	86.87(+2.54)	95.55(-1.41)
		UDA	59.89(+7.72)	78.22(+4.22)	85.67(+1.34)	97.32(+0.36)
		CBERT	58.94(+6.77)	77.87(+3.87)	85.41(+1.08)	96.31(-0.65)
		AugSBERT	61.31(+9.14)	77.43(+3.43)	87.04(+2.71)	97.15(+0.19)
	HMLM	BERT-base	61.55(+9.38)	79.18(+**5.18**)	87.25(+2.92)	97.32(+0.36)
		ERNIE 2.0	60.33(+8.16)	78.45(+4.45)	87.03(+2.70)	96.89(-0.07)
		BERT-wmm	61.67(+9.50)	78.88(+4.88)	87.39(+3.06)	97.20(+0.24)
		RoBERTa	63.39(+**11.22**)	78.95(+4.95)	88.61(+**4.28**)	97.56(+**0.60**)
γ=2	Baseline	EDA	65.01(+12.84)	79.20(+5.20)	87.22(+2.89)	95.83(-1.13)
		HDA	64.13(+11.96)	78.77(+4.77)	87.36(+3.03)	95.03(-1.93)
		UDA	66.19(+14.02)	80.13(+6.13)	87.89(+3.56)	96.32(-0.64)
		CBERT	65.08(+12.91)	78.93(+4.93)	87.19(+2.86)	96.06(-0.90)
		AugSBERT	67.79(+15.62)	82.56(+8.56)	88.47(+4.14)	97.52(+0.56)
	HMLM	BERT-base	68.26(+**16.09**)	83.56(+**9.56**)	89.91(+5.58)	98.29(+1.33)
		ERNIE 2.0	68.03(+15.86)	82.25(+8.25)	88.52(+4.19)	97.44(+0.48)
		BERT-wmm	69.02(+16.85)	83.08(+9.08)	89.21(+4.88)	98.34(+1.38)
		RoBERTa	68.11(+15.94)	83.46(+9.46)	90.35(+**6.02**)	98.91(+**1.95**)
γ=3	Baseline	EDA	68.27(+16.10)	80.38(+6.38)	87.09(+2.76)	95.71(-1.25)
		HDA	67.38(+15.21)	79.02(+5.02)	86.21(+1.88)	95.21(-1.75)
		UDA	68.71(+16.54)	81.05(+7.05)	87.23(+2.90)	96.58(-0.38)
		CBERT	68.11(+15.94)	79.33(+5.33)	87.11(+2.78)	95.42(-1.54)
		AugSBERT	70.92(+18.75)	85.13(+7.7)	90.55(+6.22)	97.26(+0.3)
	HMLM	BERT-base	71.32(+19.15)	85.21(+**11.21**)	92.08(+7.75)	98.60(+1.64)
		ERNIE 2.0	70.95(+18.78)	84.69(+10.69)	90.47(+6.14)	97.32(+0.36)
		BERT-wmm	72.91(+20.74)	85.00(+11.00)	91.33(+7.00)	98.71(+1.75)
		RoBERTa	73.47(+**21.30**)	85.11(+11.11)	92.79(+**8.46**)	99.05(+**2.09**)

Accuracy(%) of HMLM vs. baselines over all datasets. Significant improvement for BERT-base and RoBERTa. Compared with non-augmented, "+" is used to indicate increase and "-" is used to indicate decrease. The results in the table are obtained by taking the mean of the three times

Bold value indicate the maximum accuracy that can be improved of all the models

the word vector and attention mechanism to determine the importance of every word to the label. There is an attention weight between each word and the label. The higher the weight, the greater the importance of the word to the label, the greater the probability of the word as a core word. For the parameters of BiLSTM+Attention, we set lr=0.001, batch_size=128, epochs=10, dropout=0.5, max_length=32, hinder_layers=128, embedding_size[1]=200.

5 Result and discussion

5.1 Results of intent detection

The final experimental results of intent detection are shown in Fig. 4, and the accuracy of the classifier improved by data augmentation is shown in Table 2. It can be seen from Fig. 4 that our method improves the accuracy of the classifier better than the baseline performance. It can also be seen from Table 2, both RoBERTa and BERT-base have the best performance, and BERT-wmm is somewhere in between. Besides, when using 10% data to expand 3 times, the accuracy of the classifier has increased by at least 0.2; using 30% data and 50% data to expand three times, the performance of the classifier is improved by about 0.1; when using the full data to increase three times, the performance of the classifier is only increased by 0.02. Such a situation indicates that the data augmentation effect is better under a small amount of data, but when the amount of data is relatively large, the change of the augmentation multiples more little effect on the classifier. However, all baselines except AugSBERT degrade classifier performance when using the full amount

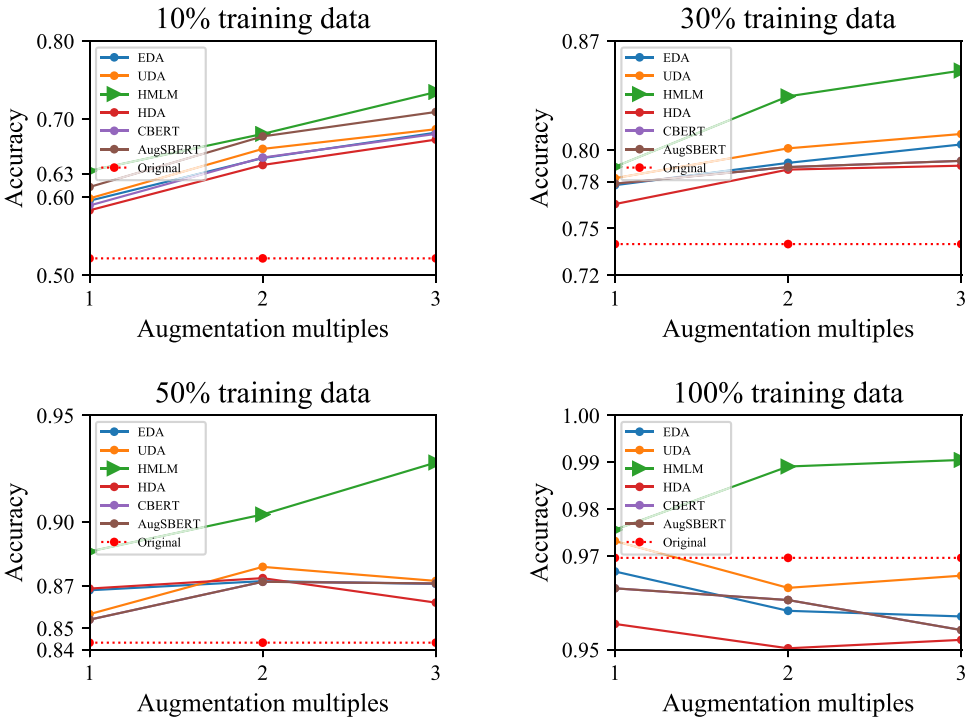

Fig. 4 The best model of HMLM and baselines respectively improve the accuracy on the intention detection task, where "Original" means to use source data without data augmentation

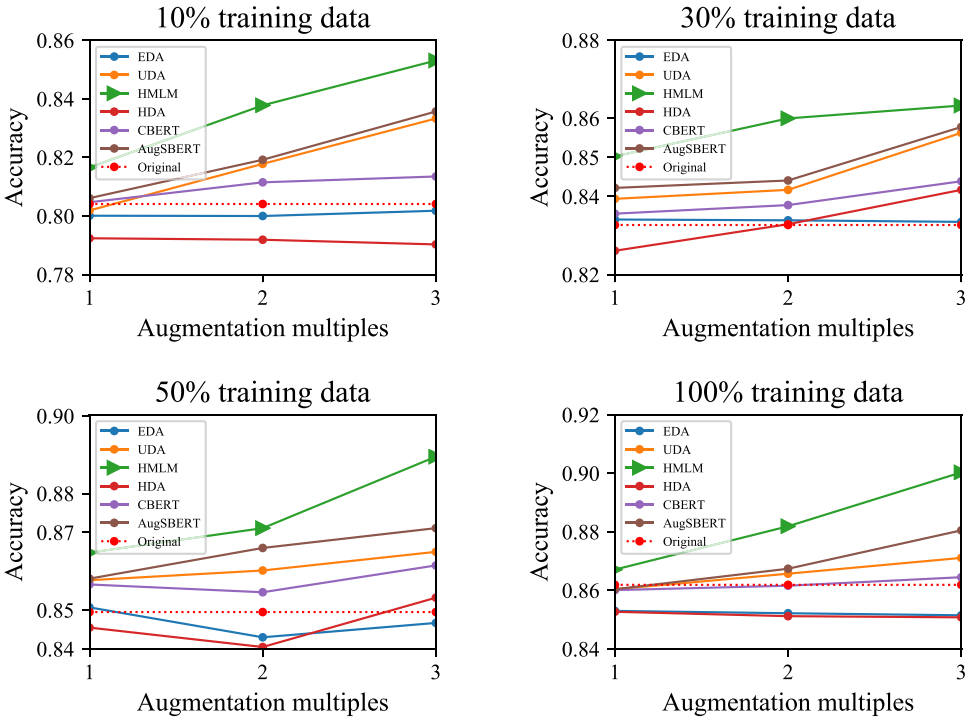

Fig. 5 The best model of HMLM and baselines respectively improve the accuracy on the news classification task, where "Original" means to use source data without data augmentation

of data to get three times as much data. This shows that they may replace the intent words related to the classification in the process of randomly selecting alternative words, which cannot guarantee the correctness of the label.

5.2 Results of news classification

The final experimental results are shown in the Fig. 5. The accuracy of the TextCNN classifier is improved by data augmentation compared to the classifier trained with the original data.

Table 3 Comparison of accuracy improvement of news classification

Multiple	Model		Training data proportion			
			β=10%	β=30%	β=50%	β=100%
γ=0	Original		80.41	83.27	84.95	86.19
γ=1	Baseline	EDA	80.01(-0.40)	83.41(+0.14)	85.07(+0.12)	85.30(-0.89)
		HDA	79.24(-1.17)	82.61(-0.66)	84.55(-0.40)	85.27(-0.92)
		UDA	80.19(-0.22)	83.94(+0.67)	85.77(+0.82)	86.04(-0.15)
		CBERT	80.47(+0.06)	83.56(+0.29)	85.66(+0.71)	86.01(-0.18)
		AugSBERT	80.61(+0.2)	84.22(+0.95)	85.81(+0.86)	86.04(-0.15)
	HMLM	BERT-base	81.52(+1.11)	84.22(+0.95)	86.32(+1.37)	86.10(-0.09)
		ERNIE 2.0	80.15(-0.26)	83.88(+0.61)	85.11(+0.16)	86.00(-0.19)
		BERT-wmm	80.78(+0.37)	83.33(+0.06)	85.56(+0.61)	86.22(+0.03)
		RoBERTa	81.66(+**1.25**)	85.03(+**1.76**)	86.48(+**1.53**)	86.71(+**0.52**)
γ=2	Baseline	EDA	80.00(-0.41)	83.39(+0.12)	84.30(-0.65)	85.22(-0.97)
		HDA	79.19(-1.22)	83.29(+0.02)	84.05(-0.90)	85.12(-1.07)
		UDA	81.78(+1.37)	84.17(+0.90)	86.02(+1.07)	86.57(+0.38)
		CBERT	81.15(+0.74)	83.78(+0.51)	85.46(+0.51)	86.16(-0.03)
		AugSBERT	81.92(+1.51)	84.41(+1.14)	86.60(+1.65)	86.74(+0.55)
	HMLM	BERT-base	82.33(+1.92)	85.63(+2.36)	87.05(+2.10)	87.55(+1.36)
		ERNIE 2.0	81.93(+1.52)	84.56(+1.29)	86.13(+1.18)	87.13(+0.94)
		BERT-wmm	82.55(+2.14)	84.67(+1.40)	86.34(+1.39)	87.03(+0.84)
		RoBERTa	83.78(+**3.37**)	86.00(+**2.73**)	87.11(+**2.16**)	88.19(+**2.00**)
γ=3	Baseline	EDA	80.18(-0.23)	83.35(+0.08)	84.67(-0.28)	85.15(-1.04)
		HDA	79.03(-1.38)	83.17(-0.10)	85.32(+0.37)	85.08(-1.11)
		UDA	83.33(+2.92)	85.63(+2.36)	86.50(+1.55)	87.11(+0.92)
		CBERT	81.35(+0.94)	84.39(+1.12)	86.15(+1.20)	86.45(+0.26)
		AugSBERT	83.57(+3.16)	85.78(+2.51)	87.11(+2.16)	88.05(+1.86)
	HMLM	BERT-base	83.78(+3.37)	85.92(+2.65)	88.71(+3.76)	89.20(+3.01)
		ERNIE 2.0	82.56(+2.15)	85.66(+2.39)	87.52(+2.57)	88.67(+2.48)
		BERT-wmm	83.01(+2.60)	85.87(+2.60)	87.15(+2.20)	88.50(+2.31)
		RoBERTa	85.31(+**4.90**)	86.33(+**3.06**)	88.95(+**4.00**)	90.03(+**3.84**)

Accuracy(%) of improvement of HMLM vs. baselines over all datasets. Significant improvement for RoBERTa. Compared with non-augmented, "+" is used to indicate increase and "-" is used to indicate decrease. The results in the table are obtained by taking the mean of the three times

Bold value indicate the maximum accuracy that can be improved of all the models

As demonstrated in Fig. 5, of all the baselines, AugSBERT achieved the best performance. At the same time, it has been found that using data augmentation sometimes reduces the performance of the classifier. In particular, EDA makes this decline more pronounced. AugSBERT randomly selects two sentences that usually lead to a dissimilar (negative) pair; positive pairs are extremely rare. This skews the label distribution of the dataset heavily towards negative pairs. We can see that in all data ratios, the proposed method is better than baselines, and the BERT-base and RoBERTa model wins with an absolute advantage. As can be seen as Table 3, RoBERTa always has stable performance under different augmentation multiples, and the gap between it and other models becomes more obvious as the dataset increases and the augmentation multiples increase. By observing the improvement of the accuracy of the classifier, it can be found that when the classifier uses 10% data increased by two times, it is better than using 30% of the original data. The same result appears in 50% of the data. This may imply that, under certain circumstances, the data obtained using data augmentation methods have more training significance than the equivalent amount of data obtained in the real world.

5.3 Results of sentiment analysis

The experimental results are shown in Fig. 6, and the accuracy of the classifier improved by data augmentation is shown in Table 4.

It can be seen from Fig. 6 and Table 4 that all the data obtained after using the augmentation model have improved the accuracy of the classifier compared to the unaugmented. Under the requirement of doubling the

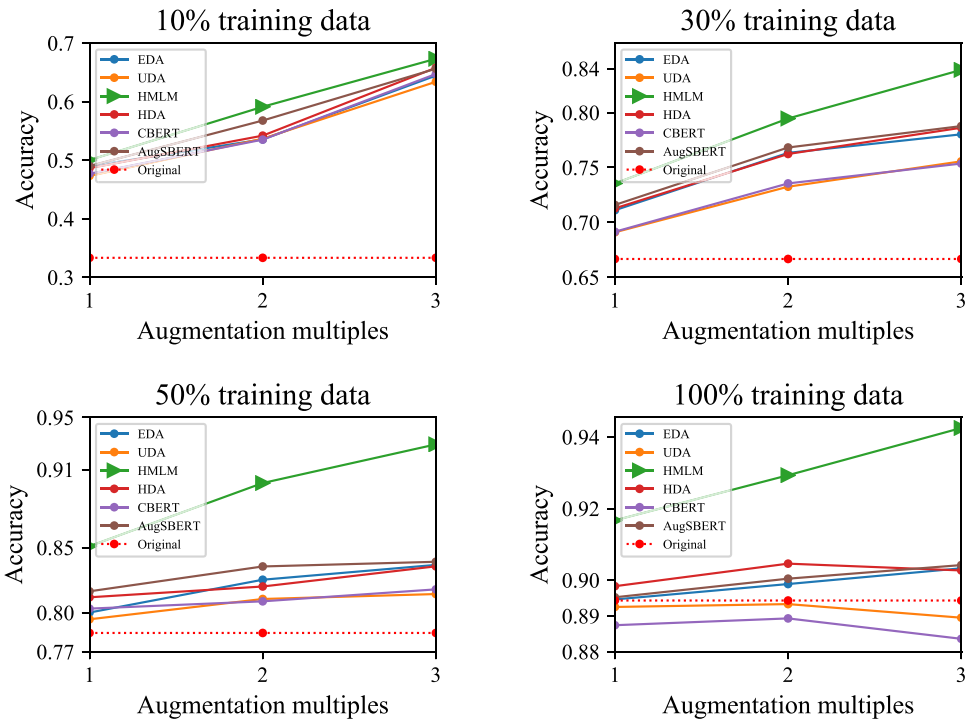

Fig. 6 The best model of HMLM and baselines respectively improve the accuracy on the sentiment analysis, where "Original" means to use source data without data augmentation

augmentation, all of the augmentation models surpass baseline methods with a weak advantage, but this transcendence is amplified with the increasing of augmentation multiples. HDA performed better in sentiment analysis than the other two tasks. We argue that it is probably because there is an explicit relationship between the label and the sentiment in the sentence. Attention is also very good at identifying this kind of relationship. When there is an implicit relationship between label and text sequence, such as news classification, attention tend to select important words are more likely to make the text sequence lose its original semantic meaning. As a result, augmented data become noise for classifier. Besides, when the augmentation multiple is triple, the RoBERTa model makes the best performance. In the same situation, with the gradual increase in the data selection rate, the performance is significantly improved. Meanwhile, we also observed that the effect of using 10% to augment the two times is equivalent to directly using 30% of the original data for classification. When using 50% of the data, it needs to be augmented by three times to achieve the effect of classification under the total amount of data. This situation shows that for sentiment analysis, data augmentation has a higher cost-performance ratio when the amount of data is relatively small.

5.4 Analysis of β and γ

From the above experimental results, the augmentation benefits of the three tasks all follow the same trend. With the β keeping the invariant, the increase in γ improves the augmentation gain. With the γ keeping the invariant, the increase in β decreases the augmentation gain. The smaller the β, the more significant the data augmentation gain, which can lead to an increase of up to 30 percentage points. For the entire data condition β=100%, the data augmentation may generate noise or even cause a negative gain, but this situation disappears with the increase of gamma.

5.5 Ablation study

So far, empirical outcomes have been favorable. In this section, we perform an ablation study to investigate the effect of the CWR procedure in HMLM. We can assume that the core word recognition will greatly increase the gain of the HLML algorithm; therefore, we isolate each CWR operation to determine its ability to boost performance. For three NLP tasks, we ran models using HMLM and HMLM w/o CWR separately on same augmentation multiples and data selection ratio. The detailed results are shown in Table 5. We find that with the same data selection ratio, merely utilizing HMLM w/o CWR can also improve the performance of the classifier, while the HMLM can improve more.

The experimental results reveal that the gain of the CWR has some discrepancies in different tasks. As shown in Table 5, for intent recognition and sentiment analysis, we received the best gain, indicating that the CWR method is suitable for these two tasks. Besides, when employing 10% and 30% data for data augmentation, the

Table 4 Comparison of accuracy improvement of sentiment analysis

Multiple	Model		Training data proportion			
			β=10%	β=30%	β=50%	β=100%
γ=0	Original		33.33	66.67	78.44	89.43
γ=1	Baseline	EDA	48.96(+15.63)	71.11(+4.44)	80.01(+1.57)	89.46(+0.03)
		HDA	48.64(+15.31)	71.28(+4.61)	81.18(+2.74)	89.83(+0.40)
		UDA	47.32(+13.99)	69.10(+2.43)	79.49(+1.05)	89.25(-0.18)
		CBERT	47.68(+14.35)	69.15(+2.48)	80.31(+1.87)	88.74(-0.69)
		AugSBERT	49.01(+15.68)	71.59(+4.92)	81.64(+3.20)	89.52(+0.09)
	HMLM	BERT-base	50.22(+**16.89**)	73.32(+6.65)	84.48(+6.04)	91.01(+1.58)
		ERNIE 2.0	48.73(+15.40)	83.88(+5.79)	83.73(+5.29)	91.22(+1.79)
		BERT-wmm	49.21(+15.88)	72.46(+6.25)	84.15(+5.71)	91.31(+1.88)
		RoBERTa	50.03(+16.70)	73.55(+**6.88**)	85.11(+**6.67**)	91.67(+**2.24**)
γ=2	Baseline	EDA	53.55(+20.22)	76.33(+9.66)	82.53(+4.09)	89.89(+0.46)
		HDA	54.19(+20.86)	76.23(+9.56)	82.01(+3.57)	90.46(+1.03)
		UDA	53.67(+20.34)	73.25(+6.58)	81.05(+2.61)	89.33(-0.10)
		CBERT	53.51(+20.18)	73.55(+6.88)	80.87(+2.43)	88.93(-0.50)
		AugSBERT	56.77(+23.44)	76.82(+10.15)	83.55(+5.11)	90.04(+0.61)
	HMLM	BERT-base	58.63(+25.30)	79.73(+**13.06**)	90.15(+**11.71**)	92.56(+3.13)
		ERNIE 2.0	56.38(+23.05)	77.97(+11.30)	88.37(+9.93)	91.63(+2.20)
		BERT-wmm	58.66(+25.33)	78.01(+11.34)	89.21(+10.77)	92.37(+2.94)
		RoBERTa	59.12(+**25.79**)	79.46(+12.79)	89.96(+11.52)	92.93(+**3.50**)
γ=3	Baseline	EDA	64.44(+31.11)	78.02(+11.35)	83.67(+5.23)	90.33(+0.90)
		HDA	65.78(+32.45)	78.62(+11.95)	83.56(+5.12)	90.27(+0.84)
		UDA	63.39(+30.06)	75.56(+8.89)	81.43(+2.99)	88.95(-0.48)
		CBERT	64.72(+31.39)	75.36(+8.69)	81.79(+3.35)	88.36(-1.07)
		AugSBERT	65.62(+32.29)	78.77(+12.10)	83.91(+5.47)	90.42(+0.99)
	HMLM	BERT-base	66.97(+33.64)	82.58(+15.91)	92.32(+13.88)	93.87(+4.44)
		ERNIE 2.0	65.15(+31.82)	80.83(+14.16)	90.67(+12.23)	92.11(+2.68)
		BERT-wmm	66.38(+33.05)	81.25(14.58)	91.53(+13.09)	92.73(+3.30)
		RoBERTa	67.32(+**33.99**)	83.89(+**17.22**)	92.93(+**14.49**)	94.25(+**4.82**)

Accuracy(%) of improvement of HMLM vs. baselines over all datasets. Significant improvement for BERT-base and RoBERTa. Compared with non-augmented, "+" is used to indicate increase and "-" is used to indicate decrease. The results in the table are obtained by taking the mean of the three times

Bold value indicate the maximum accuracy that can be improved of all the models

effect of CWR decreases with the increase of augmentation multiples. When more data are given, the effect of CWR begins to increase. It turns out that CWR operation contributes more to performance gain for intent recognition, news classification, and sentiment analysis, and the CWR method for news classification still has room for improvement.

5.6 The situation of conserving true labels

In text data augmentation, the key is to maintain class labels while modifying the input data. However, if sentences are significantly changed, then original class labels may no longer be valid. We take a sentence-matching approach to examine whether HMLM significantly changes the meanings of augmented sentences. We performed the experiment when γ=3 and β=100% and conducted mean pooling after using tencent embedding for original sentences and augmented sentences so that we could calculate the cosine similarity of the sentence pair. We have counted the data distribution of the semantic similarity, and the result is demonstrated in Fig. 7.

We found that the original data representations and augmented sentences has highly similarity those of the original sentences, which suggests that, for the most part, sentences augmented with HMLM conserved the labels of their original sentences.

Table 5 We employ the the most stable pre-trained model (RoBERTa) above for ablation experiments

Task	Method	β=10%			β=30%			β=50%			β=100%		
		γ=1	γ=2	γ=3	γ=1	γ=2	γ=3	γ=1	γ=2	γ=3	γ=1	γ=2	γ=3
Intent detection	HMLM	63.39	69.02	73.47	79.18	83.56	85.21	88.61	90.35	92.79	97.56	98.91	99.05
	HMLM w/o CWR	53.25	55.79	58.96	74.15	76.01	76.83	84.25	85.03	85.41	95.52	95.31	95.20
News classification	HMLM	81.66	83.78	85.31	85.03	86.00	86.33	86.48	87.11	88.95	86.71	88.19	90.03
	HMLM w/o CWR	80.52	82.63	84.41	83.55	84.43	85.17	85.52	85.86	86.99	86.03	87.15	88.62
Sentiment Analysis	HMLM	50.22	59.12	67.32	73.55	79.73	83.89	85.11	90.15	92.93	91.67	92.93	94.25
	HMLM w/o CWR	35.61	38.29	40.56	67.23	68.63	70.11	78.94	78.12	77.33	88.32	87.85	86.19

Average performance gain of HMLM and HMLM w/o CWR operations over three NLP tasks for different training set sizes

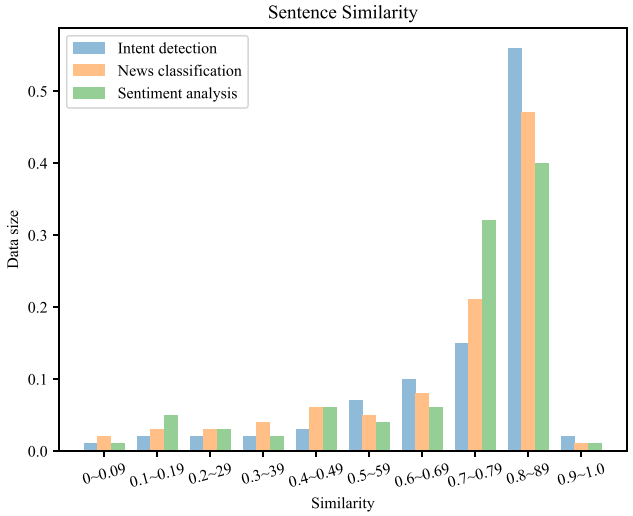

Fig. 7 Semantic similarity calculation of original sentences and augmented sentences. We performed the experiment when γ=3 and β =100%, and conducted mean pooling after using tencent embedding to calculate the cosine similarity. The higher the similarity between the augmented sentence and the original sentence, the more consistent the linguistic space is, suggesting that augmented sentences maintained their true class labels

6 Conclusion

In the present work, we introduce a novel augmentation method by heuristic Masked Language Modeling. The proposed method explores the selection of core words for various tasks, after which non-core words are masked, and the masked fragments are predicted using the pre-trained language model. It provides a priori knowledge for data augmentation, which not only maintains label consistency but also enriches semantics. We have conducted experiments on intent recognition, news classification, and sentiment analysis, which demonstrated that our method could generate a variety of words appropriately with the semantic tags of the original text and improve the neural classifier more than the baseline. This method is simple, effective, and easy to implement, providing insight for practitioners and researchers to select use cases in data-starved research and applications. In future works, we will explore mixed augmentation methods based on more pre-trained models. Also, we will explore the application of our augmentation method for other natural language processing tasks to make continued progress in text data augmentation.

References

1. Xie Q, Dai Z, Hovy E.H, Luong T, Le Q (2020) Unsupervised data augmentation for consistency training. In: Advances in neural information processing systems 33: annual conference on neural

information processing systems 2020, NeurIPS 2020, December 6-12,
2. Anaby-Tavor A, Carmeli B, Goldbraich E, Kantor A, Kour G, Shlomov S, Tepper N, Zwerdling N (2020) Do not have enough data? deep learning to the rescue! In: The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, New York, NY, USA, February 7-12, 2020, pp. 7383–7390
3. Wang J, Yang Y, Liu K, Xie P, Liu X (2022) Instance-guided multi-modal fake news detection with dynamic intra- and inter-modality fusion. In: Advances in knowledge discovery and data mining—26th Pacific-Asia conference, PAKDD 2022, Chengdu, China, May 16-19, 2022, pp. 510–521
4. Liu K, Li T, Yang X, Yang X, Liu D, Zhang P (2022) Wang J Granular cabin: an efficient solution to neighborhood learning in big data. Inform Sci 583:189–201
5. Tobin J, Fong R, Ray A, Schneider J, Zaremba W, Abbeel P (2017) Domain randomization for transferring deep neural networks from simulation to the real world. In: 2017 IEEE/RSJ International conference on intelligent robots and systems, IROS 2017, Vancouver, BC, Canada, September 24-28, 2017, pp. 23–30
6. Hoang C.D.V, Koehn P, Haffari G, Cohn T (2018) Iterative back-translation for neural machine translation. In: Proceedings of the 2nd workshop on neural machine translation and generation, NMT@ACL 2018, Melbourne, Australia, July 20, 2018, pp. 18–24
7. Edunov S, Ott M, Auli M, Grangier D (2018) Understanding back-translation at scale. In: Proceedings of the 2018 conference on empirical methods in natural language processing, Brussels, Belgium, October 31 - November 4, 2018, pp. 489–500
8. Fadaee M, Bisazza A, Monz C (2017) Data augmentation for low-resource neural machine translation. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, pp. 567–573
9. Kobayashi S (2018) Contextual augmentation: Data augmentation by words with paradigmatic relations. In: Proceedings of the 2018 conference of the North American chapter of the association for computational linguistics: human language technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, pp. 452–457
10. Wu X, Lv S, Zang L, Han J, Hu S (2019) Conditional bert contextual augmentation. In: Computational Science—ICCS 2019—19th International Conference, Faro, Portugal, June 12-14, 2019, pp. 84–95
11. Liu T, Cui Y, Yin Q, Zhang W, Wang S, Hu G (2017) Generating and exploiting large-scale pseudo training data for zero pronoun resolution. In: Proceedings of the 55th annual meeting of the association for computational linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, pp. 102–111
12. Hou Y, Liu Y, Che W, Liu T (2018) Sequence-to-sequence data augmentation for dialogue language understanding. In: Proceedings of the 27th international conference on computational linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018, pp. 1234–1245
13. Dong L, Mallinson J, Reddy S, Lapata M (2017) Learning to paraphrase for question answering. In: Proceedings of the 2017 conference on empirical methods in natural language processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017, pp. 875–886
14. Wei JW, Zou K (2019) EDA: Easy data augmentation techniques for boosting performance on text classification tasks. 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 2019, Hong Kong, China, November 3-7, 2019, pp. 6382–6388
15. Dai X, Adel H (2020) An analysis of simple data augmentation for named entity recognition. In: Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020, pp. 3861–3867
16. Vania C, Kementchedjhieva Y, Søgaard A, Lopez A (2019) A systematic comparison of methods for low-resource dependency parsing on genuinely low-resource languages. In: Inui, K., Jiang, J., Ng, V., Wan, X. (eds.) Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019, pp. 1105–1116
17. Gulordava K, Bojanowski P, Grave E, Linzen T, Baroni M Colorless green recurrent networks dream hierarchically. In: Proceedings of the 2018 conference of the North American chapter of the association for computational linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, pp. 1195–1205
18. Sennrich R, Haddow B, Birch A Edinburgh neural machine translation systems for WMT 16. In: Proceedings of the first conference on machine translation, WMT 2016, colocated with ACL 2016, August 11-12, Berlin, Germany, pp. 371–376
19. Gal Y, Ghahramani Z A theoretically grounded application of dropout in recurrent neural networks. In: Advances in neural information processing systems 29: annual conference on neural information processing systems 2016, December 5-10, 2016, pp. 1019–1027
20. Norouzi M, Bengio S, Chen Z, Jaitly N, Schuster M, Wu Y, Schuurmans D Reward augmented maximum likelihood for neural structured prediction. In: Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain, pp. 1723–1731
21. Sennrich R, Haddow B, Birch A Improving neural machine translation models with monolingual data. In: Proceedings of the 54th annual meeting of the association for computational linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, pp. 86–96
22. Mallinson J, Sennrich R, Lapata M Paraphrasing revisited with neural machine translation. In: Proceedings of the 15th conference of the European chapter of the association for computational linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, pp. 881–893
23. Yu A.W, Dohan D, Luong M, Zhao R, Chen K, Norouzi M, Le Q.V Qanet: Combining local convolution with global self-attention for reading comprehension. In: 6th international conference on learning representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018
24. Li Y, Cohn T, Baldwin T Robust training under linguistic adversity. In: Proceedings of the 15th Conference of the European chapter of the association for computational linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, pp. 21–27
25. Yasunaga M, Kasai J, Radev D.R Robust multilingual part-of-speech tagging via adversarial training. In: Proceedings of the 2018 conference of the North American chapter of the association for computational linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, pp. 976–986
26. Alzantot M, Sharma Y, Elgohary A, Ho B, Srivastava M.B, Chang K Generating natural language adversarial examples. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018, pp. 2890–2896
27. Qiu X, Sun T, Xu Y, Shao Y, Dai N (2020) Huang X Pre-trained models for natural language processing: a survey. Sci China Technol Sci 63:1872–1897

28. Devlin J, Chang M, Lee K, Toutanova K 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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, pp. 4171–4186
29. Sun WSLYFSTHWHWH Y Ernie 2.0: A continual pre-training framework for language understanding. In: The Thirty-fourth AAAI conference on artificial intelligence, AAAI 2020, New York, NY, USA, February 7-12, 2020, pp. 8968–8975
30. Cui Y, Che W, Liu T, Qin B, Yang Z Pre-training with whole word masking for chinese BERT. IEEE ACM Trans. Audio Speech Lang. Process. 29 3504–3514 (2021)
31. Xie Z, Huang Y, Zhu Y, Jin L, Liu Y, Xie L Aggregation cross-entropy for sequence recognition. In: IEEE conference on computer vision and pattern recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019, pp. 6538–6547
32. Taylor W.L "cloze procedure": A new tool for measuring readability. Journalism quarterly 30(4), 415–433 (1953)
33. Yu S, Yang J, Liu D, Li R, Zhang Y (2019) Zhao S Hierarchical data augmentation and the application in text classification. IEEE Access 7:185476–185485
34. Thakur N, Reimers N, Daxenberger J, Gurevych I Augmented SBERT: data augmentation method for improving bi-encoders for pairwise sentence scoring tasks. In: Proceedings of the 2021 conference of the North American chapter of the association for computational linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021, pp. 296–310
35. Kim Y Convolutional neural networks for sentence classification. In: Proceedings of the 2014 conference on empirical methods in natural language processing, EMNLP 2014, October 25-29, 2014, pp. 1746–1751
36. Mihalcea R, Tarau P Textrank: Bringing order into texts. In: Proceedings of the 2016 conference on empirical methods in natural language processing, EMNLP 2004,Barcelona, Spain, July