

High Utility Pattern Fusion by Pretrained Language Models for Text Classification

Yujia Wu, Hong Ren, Xuan Zhang, Guohua Xiao*

School of Information Science and Technology, Sanda University, Shanghai, 201209, China.

*Corresponding author(s). E-mail: guohua.xiao@gmail.com

Abstract. In the area of text classification, the identification of correlation patterns among semantics presents a persistent challenge. To tackle this issue, we propose a method called High Utility Pattern (HUP) fusion by Pretrained Language Models for Text Classification, which aims to enhance the performance of text classification techniques by learning correlation patterns among semantics within the same space. Specifically, HUP employs a Triplet Networks architecture, which utilizes three distinct encoders to extract sample semantics, correlation pattern information, and label semantic information, respectively. We employ a high-utility itemset mining algorithm to extract correlation pattern information with high utility, and by incorporating prompt templates into labels, the model is able to fully leverage the semantic knowledge embedded in pre-trained models. Ultimately, through joint training, the distance between a sample and its corresponding label is minimized, while the distance between the sample and labels that are not associated with the sample is maximized. Empirical investigations conducted on six standard text classification datasets reveal that the classification accuracy of HUP exhibits a notable enhancement, with an average accuracy increase ranging from 1.52% to 89.08%.

Keywords: Text Classification, Transformer Encoders, High Utility Pattern, Pre-trained Language Models

1 Introduction

In recent years, Pre-trained Language Models (PLMs) have demonstrated tremendous potential in text classification tasks, leveraging their linguistic knowledge [1-3]. In natural language, there exist temporal relations between events (before, after, and simultaneous). Some methods have been employed to extract temporal relations between events, which are crucial for natural language understanding [4]. However, not only do temporal precedence relations exist in natural language, but also correlation patterns [5-10].

Nevertheless, due to the limitations of input sequence length, existing Transformer-based methods struggle to learn various long-distance correlation patterns in the input sequence. Long-distance correlation patterns are important information for learning semantic features. For instance, A and B may have a strong correlation, but they appear far apart in a sentence (exceeding the length of the input sequence), making it difficult

to capture their correlation patterns using a context window. Therefore, extracting various long-distance correlation patterns in natural language for text classification tasks remains a challenge.

High Utility Pattern is a significant technique in the field of data mining [5,6,11], capable of breaking the limitations of context windows and effectively mining high-utility correlation patterns, thus capturing long-distance correlation pattern information in samples. Inspired by this, a Triplet Networks architecture is utilized, employing three distinct Transformer Encoders to extract sample features, high-utility correlation pattern features, and label semantic features. Specifically, a high-utility model is utilized to mine long-distance correlation pattern information, and label semantic features are learned by adding a prompt template to the labels. Finally, learn the distance information between the training samples and labels through joint training.

The experimental results on six publicly available text classification datasets show that the proposed HUP model outperforms directly fine-tuned PLM models. The robust performance of the HUP model is attributed to the features it captures. The contributions are threefold:

- (1) Proposing the incorporation of High Utility Pattern as a data enhancement mechanism for text classification tasks to enhance model performance;
- (2) Assimilating distance information between samples and labels through a Triplet Networks architecture;
- (3) Demonstrating through results obtained from six commonly employed text classification datasets that the proposed approach surpasses previous state-of-the-art methodologies, thereby affirming the efficacy of our approach.

2 Related Work

In the current era of booming deep learning, mainstream methods for text classification have widely adopted advanced techniques such as Convolutional Neural Networks (CNNs) [12,13]. To enrich text representations and enhance models' external knowledge, these methods are often supplemented with word embedding tools [14]. An innovative dual contrastive learning framework that significantly improves text classification performance through label-aware data augmentation has been proposed [15]. Techniques have also been developed to generate labels during the prediction process, implicitly utilizing the semantic information of labels in text classification tasks [16]. Furthermore, efforts have been made to expand the label word space, enriching the model's semantic understanding with the help of external knowledge bases [17]. Additionally, a label-semantic-aware pre-trained model has been proposed, which effectively enhances the model's performance on text classification tasks by deeply utilizing the semantic information of labels [18]. In numerous studies, labels are no longer simply encoded as numbers; instead, their rich semantic information is fully explored and utilized [19]. Improving classification accuracy by adding prompt templates to labels, allowing the model to learn the distance relationship between labels and sentences, has also been explored [20].

High Utility Itemset Mining (HUIM) is a key technique in data mining, aiming to identify high-value itemsets from databases to support decision-making. A top-k mining method based on genetic algorithms has been proposed to reduce runtime and memory consumption [5]. Utility lists have been utilized to simplify the search process [6]. Closed high utility itemsets have been mined over data streams [7], and patterns with negative unit profits have also been mined [8]. Improvements in algorithm performance through new data structures and search strategies have been developed [9], and parallel models using the MapReduce framework have been constructed [10]. This paper utilizes HUIM to mine association pattern information of words in datasets, which is not limited by distance dependencies.

3 Method

The provided text delineates a methodological approach for processing an input text sequence. Initially, the input text is passed through an Encoder, undergoing an encoding process. Concurrently, the text sequence is directed into a high utility association pattern extractor, which generates association pattern information to serve as data augmentation for the input sequence. Subsequently, the association pattern features are integrated with the features derived from the input text sequence. The corresponding labels of the input sequence are then enhanced by employing prompt templates to increase the sequence length, and these enhanced labels are passed through another Encoder for encoding. Finally, a multi-head attention mechanism is utilized to learn the features between the samples and their respective labels.

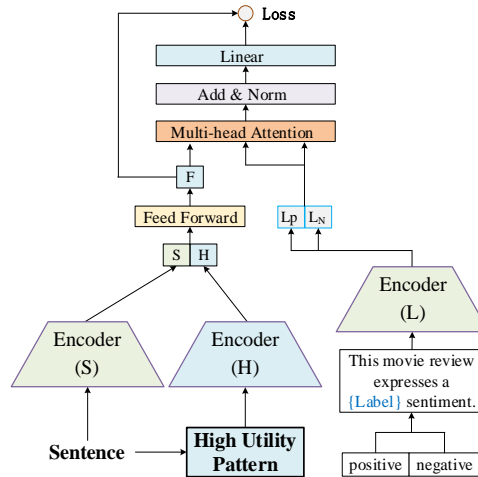


Fig. 1. The proposed framework encompasses several key components. Specifically, the distance information between the training samples and their corresponding labels is learned through a joint training process.

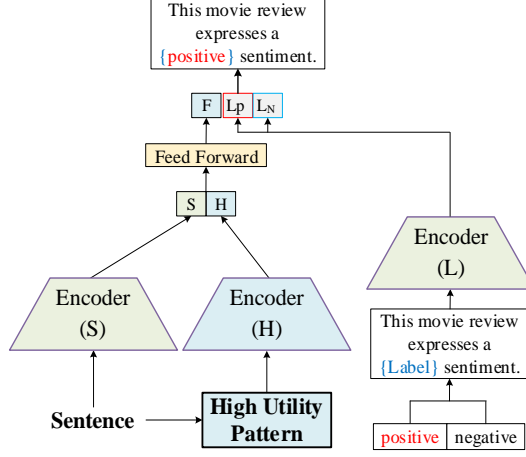


Fig. 2. The evaluation process of the proposed methodology comprises a series of systematic steps.

3.1 Formalizations

The original text sequence $D = \{i_1, i_2, \dots, i_L\}$ is considered, encompassing k categories. The text sequence D consists of L tokens, with each token $i^L \in R^N$ being denoted by an N -dimensional word embedding. Figure 1 provides an overview of the comprehensive framework of the proposed HUP. The model processes a text sequence D as input, yielding token i^L , and extracts sentence features S , high-utility pattern features H , and label features $L_j (j = 1, 2, \dots, k)$ via three distinct Encoders, respectively. Each text sequence feature is encapsulated by $[CLS]$, capturing the overall semantic information of the sentence. For each label, a prompt template is employed to extend its length, following which the encoder is utilized to derive the label feature $L_j (j = 1, 2, \dots, k)$.

Subsequently, the sentence features S and high-utility pattern features H are fused to yield fused features F . A multi-head attention layer is deployed to integrate the features F and $L_j (j = 1, 2, \dots, k)$ in order to assimilate distance information between samples and labels. During training, the model endeavors to minimize the distance between the features F and the corresponding labels $L_j (j = 1, 2, \dots, k)$ while maximizing the distance from other non-corresponding labels $L_j (j = 1, 2, \dots, k)$.

During model testing, the test samples are processed through an encoder to acquire fused features F . Simultaneously, all labels are extended with prompt templates, and the resultant sequences are channeled into another encoder to obtain label features $L_j (j = 1, 2, \dots, k)$. In the final phase, the distances between F and $L_j (j = 1, 2, \dots, k)$ are calculated, and the label associated with the smallest distance is chosen as the definitive classification outcome, as illustrated in Figure 2.

3.2 Extraction of Text features

The proposed framework comprises a text sequence encoder that initially processes the input text sequence D and retrieves the word embeddings W_E^T for the text sequence D . Subsequently, W_E^T is directed into a text encoder to derive the feature representation S of the text sequence. Various pre-trained models can be employed for feature extraction from the text sequence. Here, S denotes the text features extracted using the Encoder, which takes the input of text sequence D and computes the text feature representation through the following function:

$$S = \text{Encoder}_S(D * W_E^T) \quad (1)$$

In this context, W_E signifies the word embedding matrix, and W_E^T denotes the transpose of W_E . In this context, we utilize the hidden state of the final layer in the Encoder as the feature representation.

Concurrently, the input sequence D undergoes processing by the high-utility pattern mining layer to procure high-utility features H . Subsequently, the features S and H are concatenated and passed through an MLP layer to acquire the fused feature F , as follows:

$$F = \text{MLP}_F(S \oplus H) \quad (2)$$

Here, MLP_F denotes a linear layer utilized to reduce the dimensionality of the features. The symbol \oplus signifies the concatenation of the two features.

For different labels $C_j (j = 1, 2, \dots, k)$, a prompt template is employed to generate a set of sentences $T_j (j = 1, 2, \dots, k)$, expressed as follows:

$$T_j = \text{TEMPLATE}(C_j) \quad (3)$$

Each sentence T_j representing a label category is directed into a text encoder to derive the feature representation L_j of the label, as follows:

$$L_j = \text{Encoder}_L(T_j) \quad (4)$$

Subsequently, L_j is directed to a multi-head attention layer alongside the fused feature F . By computing attention scores, a feature Z_j representing the distance information between the sentence and different labels is derived, with the calculation method being as follows:

$$Z_j = \text{MLP}_z(\text{MultiHead}(K, Q, V)) \quad (5)$$

Here, Q is obtained by multiplying the sentence feature F with a trainable parameter sharing matrix W^Q . K and V are obtained by multiplying L_j with two trainable parameter sharing matrices W^K and W^V , respectively. MultiHead is a shorthand for multi-head attention, and MLP_z denotes a linear layer.

Z_j serves as a feature vector encompassing the distance information between the sentence and the labels. Z_j is compared with the sentence feature S to compute the distance.

3.3 High Utility Pattern

To capture the associative relationships between words, a high-utility pattern mining layer is employed to extract essential and strongly associated features for input into the encoder. In this context, each item represents a token $i^L \in R^N$, where each token i^L

appears once or multiple times in the input sequence D , and the number of occurrences is denoted as the utility value $U(i^L, D_i)$ of the word. Table 1 furnishes a specific example, presuming that the dataset encompasses four input sequences D_i , with each token i^L represented by a lowercase letter.

Table 1. Example of Document D

NO.	Sentences	Item and utility
D_1	c a c e c e	(a,1) (c,3) (e,2)
D_2	a b a f e f	(a,2) (b,1) (e,1) (f,2)
D_3	d b d f d	(b,1) (d,3) (f,1)
D_4	b d c d b e	(b,2) (c,1) (d,2) (e,1)

Definition 1: The utility $U(i^L, D_i)$ of a token i^L in a sentence D_i is defined as $U(i^L, D_i) = \text{Count}(i^L, D_i)$, where $\text{Count}(\cdot)$ denotes the counting function. For instance, in Table 1, $U(c, D_1) = 3$.

Definition 2: The utility $U(X, D_i)$ of an itemset X in a sentence D_i is defined as $U(X, D_i) = \sum_{i_k \in X \wedge X \in D_i} U(i_k, D_i)$. For example, $U(\{ac\}, D_1) = U(\{a\}, D_1) + U(\{c\}, D_1) = 4$.

Definition 3: The utility $U(X)$ of an itemset X in a dataset DB is defined as $U(X) = \sum_{X \in D_i \wedge D_i \in DB} U(X, D_i)$. For instance, $U(\{ae\}) = U(\{ae\}, D_1) + U(\{ae\}, D_2) = 6$.

Definition 4: Given a user-defined minimum utility threshold th , an itemset X is considered a high-utility itemset if $U(X) \geq th$. In Table 1, assuming $th = 4$, $\{ae\}$ is a high-utility itemset since its utility value equals 6.

Assuming a threshold th for high-utility itemsets set to 4, all high-utility 2-itemsets in the dataset DB are: $\{ae\}$, $\{af\}$, $\{bd\}$, $\{be\}$, $\{bf\}$, $\{ce\}$, $\{df\}$.

Each category's high-utility 2-itemsets compose a high-utility pattern filter Ψ_k , where Ψ_k represents the set of all high-utility 2-itemsets with utility values above the threshold th for samples belonging to category k . The specific calculation method is as follows:

$$\Psi_k = \{X | X \in DB^k, U(X) \geq th\} \quad (6)$$

Here, k denotes the category label, and DB^k signifies the set of all samples belonging to the k -th category.

Following the preliminary filtration of high-utility pattern features through the mining algorithm, these features are encoded through an encoder to derive the feature H , as follows:

$$H = \text{Encoder}_H(\Psi_k) \quad (7)$$

3.4 Loss Function

The High Utility Pattern (HUP) model processes an input text sequence D , applies word embedding to derive the text feature representation through an encoder, and subsequently directs it into an *MLP* layer for prediction. To facilitate end-to-end training, we introduce a novel joint loss function. The extracted features are computed through the function f_c , and the resulting output is directed into the final layer of the proposed model, as expressed by the equation:

$$\hat{y} = \text{softmax}(f_c \cdot W_c + b_c) \quad (8)$$

where W_c and b_c represent the weights and biases of the model, respectively.

The training objective of HUP aims to minimize the distance between the feature F and the feature vector Z_i fused with the corresponding label, as they originate from the same class of samples. Conversely, for feature vectors of labels that do not align with F , the objective is to maximize the distance. This is achieved through the construction of a loss function. The cosine similarity between the two vectors F and Z_i is calculated as follows:

$$\text{Sim}(F, Z_i) = \frac{F \cdot Z_i}{\|F\| \times \|Z_i\|} \quad (9)$$

where the range of $\text{Sim}(F, Z_i)$ spans $[-1, 1]$, with -1 indicating complete dissimilarity and 1 indicating complete similarity. The loss function is defined as follows:

$$\text{Loss} = \sum_{i=1}^k L(F, Z_i, y)^i \quad (10)$$

where the formula for $L(F, Z_i, y)$ is given as:

$$L(F, Z_i, y) = \frac{(1-y)}{2} (\text{Sim}(F, Z_i) + 1)^2 + \frac{y}{2} (\text{Sim}(F, Z_i) - 1)^2 \quad (11)$$

Here, y represents the sample label, with a value of 1 for matching samples and 0 for non-matching samples.

The proposed framework aims to optimize the distance between the feature vectors generated by the sample and its corresponding label, while simultaneously maximizing the distance to feature vectors generated by non-corresponding labels. This approach seeks to enhance the model's ability to effectively differentiate between the target labels and improve the overall classification performance.

4 Experiment

4.1 Datasets and Baselines

To evaluate the proposed method, experiments were conducted on six publicly available datasets, including MPQA [21], SUBJ [22], TREC [23], MR [24], SST1 [25], and SST2 [25].

Table 2. Dataset Statistics and Prompt Templates

Dataset	class	Avg.L	size
MR	2	20	10662
MPQA	2	3	10604
SUBJ	2	23	9999
SST1	5	18	11855
SST2	2	19	9613
TREC	6	10	5891

To fully demonstrate the performance of the proposed model, several industry-leading benchmark algorithms were specifically selected for detailed comparison, including BERT [26], ELECTRA [27], RoBERTa [28], and Muppet [29].

4.2 Experimental Results

Table 3 presents the detailed experimental results. During the experiments, various pre-trained models were employed, including Bert-base, ELECTRA-base, RoBERTa-base, and Muppet-RoBERTa-base, which were fine-tuned to optimize the model performance. Accuracy was chosen as the primary performance metric. The notation "HUP+X" denotes the use of X as the sentence feature extractor, and in this experiment, Muppet-RoBERTa-base was selected as the high-utility pattern feature extractor and label feature extractor.

Table 3. The proposed method and baseline methods were evaluated on six text classification datasets.

Dataset	MR	MPQA	SUBJ	SST1	SST2	TREC
<i>BERT</i>	87.45±0.18	91.06±0.18	96.48±0.26	52.49±0.13	93.55±0.17	94.63±0.64
HUP+BERT	88.89±0.48	91.74±0.49	96.88±0.19	55.76±0.50	96.80±0.55	96.13±0.22
<i>ELECTRA</i>	90.24±0.15	91.23±0.15	97.12±0.30	55.19±0.13	96.77±0.16	95.72±0.53
HUP+ELECTRA	91.11±0.25	92.34±0.12	97.34±0.11	59.88±0.05	97.48±0.20	97.56±0.19
<i>RoBERTa</i>	89.81±0.11	91.36±0.20	97.16±0.21	54.26±0.13	94.80±0.17	95.59±0.27
HUP+RoBERTa	90.53±0.61	92.74±0.15	97.30±0.26	58.57±0.27	97.59±0.20	97.22±0.35
<i>Muppet</i>	94.41±0.22	92.83±0.15	97.12±0.40	59.51±0.19	95.38±0.12	96.26±1.05
HUP +Muppet	94.84±0.15	93.77±0.15	97.70±0.16	59.81±0.19	97.81±0.16	98.06±0.19

The observations from the results presented in Table 2 are as follows. On five datasets, including MR, MPQA, SUBJ, SST2, and TREC, the proposed method achieved the optimal accuracy by using Muppet-RoBERTa-base as the word feature extractor. On the SST1 dataset, the use of ELECTRA-base as the sentence feature extractor achieved the best performance. The proposed method can be improved based on various pre-trained models, continuously enhancing the results of the baseline feature extractors. On these six datasets, the average accuracy increased from 87.52% to 89.08%, while the standard deviation remained unchanged at 0.26. This validates the effectiveness of the proposed model, demonstrating that learning the distance information between samples and labels can improve the accuracy of text classification.

4.3 Ablation Experiments and Analysis

In order to substantiate the efficacy of the HUP method delineated in this paper, we conducted ablation experiments focusing on high-utility pattern feature extraction. Given the widespread utilization of prompt templates in large language models, this subsection adopts prompt templates for label semantic feature extraction. The objective

is to ascertain whether the incorporation of high-utility pattern features as data augmentation for samples can yield enhancements in model performance. For the ablation experiments, we selected the six datasets featured in this paper, as detailed in Table 4.

Table 4. The comparison between the proposed method and baseline methods was conducted across six text classification datasets, incorporating the addition of high-utility pattern features.

Dataset	BERT	ELECTRA	RoBERTa	Muppet
MR	87.71±0.15	90.94±0.16	90.06±0.48	94.62±0.11
+ HUP	88.89±0.48	91.11±0.25	90.53±0.61	94.84±0.15
MPQA	91.57±0.22	92.26±0.15	91.68±0.18	93.55±0.11
+ HUP	91.74±0.49	92.34±0.12	92.74±0.15	93.77±0.1
SUBJ	96.70±0.16	97.28±0.19	97.26±0.27	97.54±0.37
+ HUP	96.88±0.19	97.34±0.11	97.30±0.26	97.70±0.16
SST1	54.62±0.49	57.07±0.13	57.38±0.46	59.68±0.4
+ HUP	55.76±0.50	59.88±0.05	58.57±0.27	59.81±0.19
SST2	96.02±0.20	97.07±0.27	96.84±0.22	96.61±0.19
+ HUP	96.80±0.55	97.48±0.20	97.59±0.20	97.81±0.16
TREC	95.25±0.27	97.15±0.47	96.47±0.41	97.15±0.33
+ HUP	96.13±0.22	97.56±0.19	97.22±0.35	98.06±0.19

In Table 4, we present a comparative analysis of the baseline method with the incorporation of high-utility pattern features across six datasets. The experimental findings unequivocally demonstrate that the inclusion of high-utility pattern features yields a substantial performance enhancement across all datasets. Here, the notation +HUP signifies the addition of high-utility pattern features, with Muppet-RoBERTa-base specifically chosen as the extractor for high-utility pattern features. Accuracy (%) serves as the evaluation metric, and each entry in the table represents the average accuracy derived from five experiments, accompanied by the standard deviation to ensure the stability and reliability of the results. In this context, +HUP indicates the retention of high-utility pattern features from HUP, while the absence of +HUP denotes the exclusion of these features during training, with all other experimental conditions remaining consistent.

Through comparative experiments, we observed a consistent trend wherein the removal of high-utility features led to a notable decline in model accuracy, regardless of the pre-trained model used. For instance, considering the MR dataset, the removal of high-utility features while employing BERT as the word feature extractor resulted in a 1.18% decrease in accuracy. On average across the six datasets, the accuracy experienced a reduction of 0.72%. Similar trends were observed for other pre-trained models, each manifesting varying degrees of performance degradation. These findings unequivocally underscore the pivotal role of high-utility features in HUP. Their absence precipitates a significant decline in model performance, validating our hypothesis that the incorporation of high-utility features enables HUP to capture long-distance associative

relationships in text, thereby extracting richer semantic information and ultimately enhancing the accuracy of text classification.

Conclusions

This paper introduces a High Utility Pattern Fusion method by Pretrained Language Models for Text Classification, designed to acquire long-distance associative relationship information among words within the same space to bolster the efficacy of text classification methods. Comprehensive empirical experiments were conducted on six benchmark datasets to substantiate the effectiveness of the proposed method.

Acknowledgments

This work was Sponsored by Natural Science Foundation of Shanghai(No.22ZR1445000) and Research Foundation of Shanghai Sanda University(No. 2021BSZX07).

References

1. Yujia, W., Jing, L., Jia, W., Jun, C.: Siamese Capsule Networks with Global and Local Features for Text Classification. *Neurocomputing* 390, 88-98 (2020)
2. Yujia, W., Xin, G., Kangning, Z.: CharCaps: Character-level Text Classification using Capsule Networks. In: *International Conference on Intelligent Computing*. pp. 187-198. Springer (2023)
3. Yujia, W., Xin, G., Yi, W., Xingli, C.: ParaNet: Parallel Networks with Pre-trained Models for Text Classification. In: *International Conference on Advanced Data Mining and Applications*. pp. 121-135. Springer (2023)
4. Shuaicheng, Z., Qiang, N., Lifu, H.: Extracting Temporal Event Relation with Syntax-guided Graph Transformer. In: *Findings of the Association for Computational Linguistics: NAACL 2022*. pp. 379-390 (2022)
5. José María Luna, Uday, K.R., Philippe, F.V., Ventura Sebastián Ventura: Efficient mining of top-k high utility itemsets through genetic algorithms. *Information Sciences* 624, 529-553 (2023)
6. Zaihe, C., Wei, F., Wei, S., Chun-Wei, L.J., Bo, Y.: An efficient utility-list based high-utility itemset mining algorithm. *Applied Intelligence* 53(6),6992-7006 (2023)
7. Meng, H., Haodong, C., Ni, Z., Xiaojuan, L., Le, W.: An efficient algorithm for mining closed high utility itemsets over data streams with one dataset scan. *Knowledge and Information Systems* 65(1), 207-240 (2023)
8. Heonho, K., Taewoong, R., Chanhee, L., Hyeonmo, K., Eunchul, Y., Bay,V., Chun-Wei, L.J., Unil, Y.: Ehmin: EHMIN: Efficient approach of list based high-utility pattern mining with negative unit profits. *Expert Systems with Applications* 209, 118214 (2022)
9. Peng, W., Xinzhen, N., Philippe, F.V., Cheng, H., Bing, W.: UBP-Miner: An efficient bit based high utility itemset mining algorithm. *Knowledge-Based Systems* 248, 108865 (2022)

10. Chun-Wei, L.J., Youcef, D., Gautam, S., Yuanfa, L., S, Y.P.: Scalable Mining of High-Utility Sequential Patterns With Three-Tier MapReduce Model. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 16(3), 1-26 (2021)
11. Hai, D., Tien, H., Thong, T., Tin, T., Bac, L., Philippe, F.V.: Efficient algorithms for mining closed and maximal high utility itemsets. *Knowledge-Based Systems* 257, 109921 (2022)
12. Yujia, W., Jing, L., Chengfang, S., Jun, C.: Words in Pairs Neural Networks for Text Classification. *Chinese Journal of Electronics* 29(3), 491-500 (2020)
13. Yujia, W., Jing, L., Vincent, C., Jun, C., Zhiqian, D., Zhi, W.: Text Classification Using Triplet Capsule Networks. In: 2020 International Joint Conference on Neural Networks (IJCNN). pp. 1-7. IEEE (2020)
14. Da Costa Liliame Soares, L, O.I., Renato, F.: Text classification using embeddings: a survey. *Knowledge and Information Systems* 65(7), 2761-2803(2023)
15. Qianben, C., Richong, Z., Yaowei, Z., Yongyi, M.: Dual Contrastive Learning: Text Classification via Label-Aware Data Augmentation. *arXiv preprint arXiv:2201.08702* (2022)
16. Giovanni, P., Ben, A., Jason, K., Ma, J., Alessandro, A., Rishita, A., dos Santos Cicero Nogueira, Bing, X., Stefano, S., et al.: Structured Prediction as Translation between Augmented Natural Languages. In: ICLR 2021-9th International Conference on Learning Representations. pp. 1-26. International Conference on Learning Representations ICLR (2021)
17. Shengding, H., Ning, D., Huadong, W., Zhiyuan, L., Jingang, W., Juanzi, L., Wei, W., Maosong, S.: Knowledgeable Prompt-tuning: Incorporating Knowledge into Prompt Verbalizer for Text Classification. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 2225-2240 (2022)
18. Aaron, M., Jason, K., Salvatore, R., Saab, M., Elman, M., Yi, Z., Dan, R.: Label Semantic Aware Pre-training for Few-shot Text Classification. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 8318-8334 (2022)
19. Markus, B., Marc, K., Christian, R.: A survey on data augmentation for text classification. *ACM Computing Surveys* 55(7), 1-39 (2022)
20. Yau-Shian, W., Ta-Chung, C., Ruohong, Z., Yiming, Y.: Pesco: Prompt-enhanced Self Contrastive Learning for Zero-shot Text Classification. In: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 14897-14911 (2023)
21. Janyce, W., Theresa, W., Claire, C.: Annotating Expressions of Opinions and Emotions in Language. *Language resources and evaluation* 39, 165-210(2005)
22. L, L., B, P.: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts. In: Proceedings of ACL-04 42nd Meeting of the Association for Computational Linguistics (2004)
23. Xin, L., Dan, R.: Learning Question Classifiers. In: COLING 2002: The 19th International Conference on Computational Linguistics. pp. 1-7 (2002)
24. Lee, B.P.L.: Exploiting Class Relationships for Sentiment Categorization with Respect to Rating Scales. In: Proceedings of the Annual Meeting of the Association for Computational Linguistics. pp. 115-124 (2005)
25. Richard, S., Alex, P., Jean, W., Jason, C., D, M.C., Y, N.A., Christopher, P.: Recursive deep models for semantic compositionality over a sentiment treebank. In: Proceedings of the 2013 conference on empirical methods in natural language processing. pp. 1631-1642 (2013)
26. Chang, K.J.D.M.W., Kristina, T.L.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: Proceedings of NAACL-HLT. pp. 4171-4186 (2019)

27. Kevin, C., Minh-Thang, L., V, L.Q., D, M.C.: ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. In: International Conference on Learning Representations (2019)
28. Yinhan, L., Myle, O., Naman, G., Jingfei, D., Mandar, J., Danqi, C., Omer, L., Mike, L., Luke, Z., Veselin, S.: RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv preprint arXiv:1907.11692 (2019)
29. Armen, A., Anchit, G., Akshat, S., Xilun, C., Luke, Z., Sonal, G.: Muppet: Massive Multi-task Representations with Pre-Finetuning. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. pp. 5799-5811 (2021)