Applied Intelligence (2022) 52:10352–10368
https://doi.org/10.1007/s10489-021-02983-2

AFE-MERT: imbalanced text classification with abstract feature extraction

Murat Okkalioglu[1,2] · **Burcu Demirelli Okkalioglu**[1]

Accepted: 5 November 2021 / Published online: 13 January 2022
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021

Abstract
The class imbalance problem occurs when the distribution among classes is not balanced. This can be a problem that causes classifier models to bias toward classes with many training samples. The class imbalance problem is inherent in text classification. The abstract feature extraction method is a versatile term weighting scheme. It serves not only as a feature extractor to form a structural form from unorganized text data but also as a dimension reduction technique and classifier. In this study, we tackle the problem of class imbalance in abstract feature extraction. The proposed method utilizes relative imbalance ratio as a factor to elevate the representation of minority classes. Besides, we also integrate relevant term factors to boost the general accuracy. Experiments conducted with three different data sets, one of which is collected for this study, show that the original abstract feature extraction method indeed suffers from the class imbalance problem and the proposed methods demonstrate significant improvements in terms of $f1\text{-}micro$, $f1\text{-}macro$, and *Matthew's correlation coefficient*. The experimental results also suggest that the proposed method is a competitive classifier and term weighting scheme when compared to the well-known classifiers (KNN, SVM, and Nearest Centroid) and term weighting schemes (TF-IDF, TF-ICF, TF-ICSDF, TF-RF, TF-PROB, TF-IGM, and TF-MONO).

Keywords Class imbalance · Abstract feature extraction · Relative imbalance ratio · Relevant term · Term weighting

1 Introduction

Most of the daily routines have been already moved to online avenues. People perform various transactions online; thus, the amount of data created online has been steadily increasing [11]. Therefore, online services that can automatically retrieve related content like Google are indispensable. Even personal data that kept offline has grown so much that people are in need of such services that can perform categorization or classifications. It is clear that the volume of data kept for either personal needs or not has surpassed the human ability to handle it. Therefore, a way of automatic handling of data is a must to organize the documents for easy access.

Text classification, which is the main concern in this study, is one of such efforts. Text classification is related to the general concept of classification [1], and the task is the process of organizing unseen text documents into predefined classes according to a model generated from labeled ones. The problem of text classification can be broken into four steps: feature extraction, dimension reduction, model creation, and evaluation [16]. In broad terms, feature extraction is the method of converting unstructured text data to structured form to be used in the created model [16]. Some pre-processing such as eliminating stop-words or stemming can be applied before the feature extraction. Since feature extraction in text classification generally results in a high dimensional representation of data, usually matrix factorization techniques are employed for dimension reduction to use classification algorithms efficiently. After dimension reduction, a classification algorithm is utilized to create a model for the unseen text documents. The last step is the evaluation step to measure the performance of the algorithm.

Abstract feature extraction method [3], AFE, is one of the feature extraction methods in the text classification literature. Although its primary task is to create a structured

✉ Murat Okkalioglu
murat.okkalioglu@yalova.edu.tr

[1] Department of Comptuer Engineering, Yalova University, Yalova, Turkey

[2] Cyber Security Application and Research Center, Yalova University, Yalova, Turkey

feature vector from unstructured text, it also serves as a dimension reduction and classification technique. AFE method creates a feature vector of size $I \times 1$ for each document, where I is the number of classes in the training set. Each feature in AFE is associated with a class. Since AFE automatically decreases the number of features into the number of classes, it acts as a dimension reduction technique as well. Moreover, the values of all features always add up to 1, and the related feature value is the probability of the document being a member of the class associated with it. Such property enables AFE to be used as a classifier as well as feature extraction and dimension reduction techniques. The effectiveness of AFE in these respects is extensively studied by [3].

Our specific motivation in this study is the data or class imbalance problem. Data or class imbalance occurs when class samples have different frequencies. In such a case, some of the classes are represented with many samples, while some others are represented with few samples [13, 15, 26]. Assume that we are assigned with news classification in different genres such as sports and finance. The natural class distribution between these could differ. The training set of articles that will decide the classification model could contain many articles for sports compared to finance. Many classifiers are sensitive to class distributions and general prediction accuracy metrics bias toward the majority class [20]. Such imbalanced data sets are common in text classification problems [18, 19, 30]. Inspired by the class imbalance fact and AFE, we scrutinize AFE in case of data imbalance in this study. The contribution of the study can be summarized as follows.

1. It is shown that the AFE suffers from the class imbalance problem. We tackle the class imbalance problem by introducing a factor based on imbalance ratio to elevate the representation of minority classes. Besides, we also propose to include a supervised term weighting factor, which determines relevant terms, to boost the general accuracy as well. Our proposed methods; therefore, do not only tackle the class imbalance problem but also improve the general accuracy significantly.
2. When used as a classifier, the proposed methods achieve a decent performance in terms of evaluation metrics, which are $f1\text{-}micro$, $f1\text{-}macro$, and *Matthew's correlation coefficient*, MCC. Moreover, the proposed approaches outperform the original AFE. We also compare the proposed approach with well-known classifiers. Results are promising to argue that the proposed method is a competitive classifier.
3. Significant improvements have been also achieved when AFE is utilized as a feature extraction (usually known as term weighting) and dimension reduction method to be used in classifiers, such as K-nearest neighbors, KNN, support vector machines, SVM, and nearest centroids, NC. Our proposed method has been compared to the state-of-the-art term weighting schemes. Results suggest that our proposed method outperforms the term weighting scheme in the literature when the number of features is reduced to the number of classes by three different feature selection methods.
4. Since we had difficulty in finding imbalanced text classification data set during our research, we also collected a data set from the Reuters UK website not only to utilize in our experiments but also to show that the natural distribution of text documents is imbalanced [19]. The collected data set will be publicly available for other researchers as well.

The rest of the paper is organized as follows. Section 2 explains the related work. AFE method is described in Section 3 as preliminary. Our contributions to AFE are represented in Section 4. Experiments are reported in Sections 5, and 6 gives the concluding remarks.

2 Related work

The aim of this paper is to handle the data or class imbalance problem, where some classes are represented by more samples while some other classes are represented by few samples, in text classification by utilizing AFE. Since AFE is a method based on term weighting, in this subsection, we focus on explaining different term weighting schemes.

Text classification can be performed as following [6]:

- Document indexing: internal representation of documents
  - Term selection: selecting a subset of terms appearing in documents to be used as features for documents
  - Term weighting: assessing how much a selected term influences characteristics of the relevant document
- Classifier learning: the creation of a learner model from the training documents

Since this paper primarily focuses on term weighting schemes, we elaborate term weighting schemes in this section. Term weighting schemes are broadly categorized into two groups, unsupervised and supervised. Unsupervised schemes do not leverage class or category membership while the supervised schemes do. In following subsections, we will review existing works in both.

2.1 Unsupervised term weighting schemes

The term weighting approach has long been studied in information retrieving literature by term frequency, TF, and inverse document frequency, IDF. TF represents how frequent the related term is while IDF informs that rare terms are more determinant than frequent terms. Basically, $TF(t_k, d_j)$ is the term frequency of a term t_k in a document d_j, and $IDF(t_k, d_j)$ is the inverse document frequency, which is the number of documents, N, divided by the number of documents having t_k as a term at least once, N_{t_k}. Notice that $IDF(t_k, d_j)$ prioritizes rare terms by the inverse frequency of documents having t_k at least once. The related TF-IDF assumption is usually represented by their multiplication. After TF-IDF is calculated, weights can be obtained by a normalization. Note that a sub-linear function such as log is usually used to dampen the effect of high occurrences in both TF and IDF calculations.

Weighted inverse document frequency, WIDF, which is an extension of IDF and replaces it in TF-IDF calculation [28]. The authors state that if different texts contain a certain term, then IDF treats all of them equally without considering how many times the term occurs in them. In other words, the term frequency is neglected, instead, a binary counting is performed. They propose WIDF, which assesses the frequency of the term, t_k, in a document, d_j, rather than binary counting. Therefore, WIDF is a factor that replaces binary counting of IDF with term frequencies, TF. Modified inverse document frequency, MIDF, is defined by [7]. In MIDF, the document frequency is normalized by the sum of term frequencies. Empirical results show that it outperforms both WIDF and TF-IDF. [25] includes missing terms as a factor in TF, IDF, and TF-IDF. Their method considers the missing terms from the document and the number of documents that do not contain the term. They propose modified TF, IDF, and TF-IDF based on this assumption.

2.2 Supervised term weighting schemes

Since the aforementioned term weighting methods do not utilize class membership information of documents, they are referred to as unsupervised. [6]'s seminal paper [6] distinguishes this fact and proposes supervised term weightings. The supervised term weighting generally considers four fundamental class membership information as in Table 1 [2]. Here, A represents the number of documents that belong to c_i and have the t_k at least once. B is the number of documents that do not belong to c_i and have t_k at least once. C represents the number of documents that belong to c_i and do not have any t_k. Finally, D is the number of documents that both do not belong to c_i and have any t_k. N is the number of total documents, which $N = A + B + C + D$.

Table 1 Contingency table for supervised term weighting schemes

	c_i	$\overline{c_i}$
t_k	A	B
$\overline{t_k}$	C	D

Supervised term weighting schemes, chi-square, information gain and gain ratio, are proposed by [6]. The authors claim that these functions, which are calculated from training samples, can be better substitutes for IDF. They can be computed by utilizing fundamental class information given in Table 1. One can refer to [19] for corresponding calculations, we skipped those to keep the paper concise. They can be utilized as dimension reduction technique as well by selecting top features.

Relevance frequency, RF, factor is proposed by [17]. RF favors terms that are concentrated on the positive category. Therefore, it uses inter-class information, which is A/B. If this ratio is greater for a term, it means that that the term occurs more in the related class, so it is assigned more weight. The authors propose TF-RF that makes RF a factor of TF.

The class imbalance problem in text classification, which is also the primary focus of this study, is addressed by [19]. The scholars claim that A/B ratio should be higher for term t_k with respect to a specific category because it indicates that the related term is more relevant to the category. Likewise, given two terms t_k and t_z, they also claim that the term with higher A/C represents c_i better because the larger portion of it occurs in c_i. Based on their assumptions, they propose probability-based term weighting, TF-PB.

Wang et al. [29] argue that term distribution among classes rather than among documents should be considered. They propose inverse category frequency, ICF, which is the ratio of total number of classes, I, to the number of classes that t_k occurs at least once, $CF(t_k)$. This method combined with TF will be called TF-ICF. The authors state that the intuition behind the ICF approach is that if a term appears in fewer categories, it becomes more distinguishing for text classification. ICF also favors rare terms and handles the class imbalance problem as well. Experimental results report that the TF-ICF approach outperforms TF, IDF, TF-IDF in terms of *f1-macro* metric.

Ren and Sohrab [24] discusses TF-IDF-ICF instead of TF-ICF and proposes the inverse class-space-density frequency, ICSDF, which uses prior knowledge of the class space. Instead of directly giving low scores to the terms appearing in multiple classes, they consider class density

Table 2 Summary of term weighting schemes used in comparisons

Term weighting schme	Formula
TF-IDF	$TF(t_k, d_j) \times IDF(t_k, d_j)$
TF-IDF-ICF	$TF\text{-}IDF \times \left(1 + \log\left(\frac{I}{CF(t_k)}\right)\right)$
TF-IDF-ICSDF	$TF\text{-}IDF \times \left(1 + \log_2\left(\frac{I}{\sum_{c_i} CD(t_k)}\right)\right)$
TF-RF	$TF \times \log_2\left(2 + \frac{A}{max(B,1)}\right)$
TF-PROB	$TF \times log_2\left(1 + \frac{A}{B}\frac{A}{C}\right)$
TF-IGM	$TF \times \frac{f_{k1}}{\sum_{r=1}^{m} f_{kr} \times r}$
TF-MONO	$TF \times MONO_{global}(t_k)$

score, CD, which is the ratio of the number of documents having term, t_k, at least once and is a member of c_i to the number of documents in c_i. The class space density, CSDF, of each term is calculated by summing all CD scores over all classes. CSDF considers class densities to weight the term, and its inverse, ICSDF, is used as a multiplication factor in TF-IDF, which is called, TF-IDF-ICSDF

Chen et al. [5] claim that the inter-class distribution has a distinguishing power and measures the inter-class distribution by inverse gravity moment, IGM. In IGM, class frequencies are sorted in a descending order, and IGM is defined as the ratio of maximum class frequency to over all frequencies which are weighted by their rank (distance to the origin). Dogan and Uysal [8] improve IGM by updating the denominator part.

Dogan and Usyal [10] propose a method that utilizes maximum occurrences, MO, and non-occurrences, NO, called TF-MONO. Their method sorts the document frequencies and the class with maximum occurrences, which is MO, is separated from the rest, which are NO. Then, MO and NO ratios are calculated, which are the ratio of the corresponding frequencies to the number of total documents in MO and NO groups. Then, MO and NO ratios are multiplied to calculate MONO. The MONO ratio in the global sense is multiplied by TF to weight them by term frequencies, which is called TF-MONO.

This study aims to address the class imbalance problem in text classification through AFE. Although some studies address the same problem [19, 29], the way we approach the problem differs from these studies. Relevance indicators are utilized by [19], and [29] exploit inverse class frequencies while we propose an imbalance ratio based factor into AFE, which results in different class-specific imbalance ratio. Besides, we also utilize A/B to boost the general accuracy, which is in part used by [19] as well. Besides, we compare our method with other term weighting schemes in the literature. Therefore, Table 2 lists those term weighting schemes that we compare our method with for the interested readers.

3 Preliminaries

As mentioned earlier, AFE [3] is a powerful tool that can serve as (i) a feature extraction method (term weighting in the general sense), (ii) a dimension reduction technique, and (iii) a classifier. In this section, we will define the steps in AFE so that our contributions in the next section can be clearly given.

AFE is very similar to TF-IDF method except for TF part. In general, TF represents the number of times a term occurs in a relevant document. However, AFE utilizes class- or category-wide frequencies for each term occurring in a document. The steps of AFE can be listed as below:

1. The total number of occurrences of a term, t_k, in a class, c_i, is calculated for each term in any document

$$n(t_k, c_i) = \sum_j n(t_k, d_j), \ d_j \in c_i \tag{1}$$

2. Calculate weight, $w_{k,i}$, how much the term, t_k, affects the relevant class, c_i

$$w_{k,i} = \log\left(n(t_k, c_i)\right) \times IDF \tag{2}$$

3. Repeat for all documents to calculate the total effects of features for d_j in c_i

$$Y_{j,i} = \sum_k w_{k,i}, \ t_k \in d_j \tag{3}$$

4. Normalize to get abstract features, $AF_{j,i}$.

$$AF_{j,i} = \frac{Y_{j,i}}{\sum_i Y_{j,i}} \tag{4}$$

The method above produces a vector of I, where I is the number of classes, for each d_j. Each feature represents d_j's probability to belong to the relevant class, c_i. Since it is guaranteed that AFE outputs I number of features, it can be referred to as a dimension reduction technique as well. Due to normalization, the features sum up to exactly 1. Moreover, AFE can be utilized as a classifier model by assigning a document into a class whose feature value is the largest. That is why AFE can be also referred to as a classifier.

Although AFE [3] remains relatively unnoticed in the term weighting literature, it deserves attention due to its versatility and reported accuracy as a feature extraction and dimension reduction technique. In this study, we scrutinize AFE and propose some improvements to it. Notice that Step 2 (2) is the most critical phase of the AFE because weights are calculated. We believe that AFE can achieve better text classification results by introducing some updates on that

step. We specially focus on the class imbalance problem and propose an update in (2). Furthermore, class-specific factors can be integrated into (2) since AFE works with class-wide frequencies of a term, $n(t_k, c_i)$, not term-document frequencies, TF. In this sense, IDF alone is a metric that does not consider the class information, so we propose a metric considering the class membership information of the term. Consequently, the next section introduces the proposed methods to improve AFE in these respects.

4 Improving AFE by elevating minority classes and relevant terms

In this section, our proposed improvements to the AFE [3] method are given. Our primary concentration is to promote minority classes, ones with fewer samples, against majority classes, ones with many samples. Besides, we also aim to modify AFE with relevant terms so that it can serve as a competitive classifier in addition to its feature extraction and dimension reduction capabilities.

4.1 Abstract feature extraction with minority elevation, AFE-ME

We tackle the class imbalance problem in AFE by introducing a factor that measures the imbalance between classes. We intend to scale the $n(t_k, c_i)$ of (2) in favor of minority classes. Imbalance in a classification problem can be measured by the number of training samples each class has. Formally, the imbalance ratio, IR, is the ratio between the class with the maximum number of samples and the class with the minimum number of samples [4, 22] as given in (5)

$$IR = \frac{\max\left(|C_1|, |C_2|, ..., |C_i|, ..., |C_{I-1}|, |C_I|\right)}{\min\left(|C_1|, |C_2|, ..., |C_i|, ..., |C_{I-1}|, |C_I|\right)}, \tag{5}$$

where $|C_i|$ represents the number of samples that class c_i has. IR in (5) gives a summary statistic for the whole data set. However, our intention is to scale $n(t_k, c_i)$ in (2) to promote minority classes; therefore, we need a metric that would result in greater values for minority classes and smaller values for majority classes. Such a measure helps AFE method strengthen the minority classes against the majority classes. Therefore, we define the *relative imbalance ratio*, RIR, which is a metric that can be computed for each class. RIR is given in (6)

$$RIR_i = \frac{\max\left(|C_1|, |C_2|, ..., |C_i|, ..., |C_{I-1}|, |C_I|\right)}{|C_i|}, \tag{6}$$

where RIR_i is RIR for c_i, and it measures the imbalance ratio between the relevant class, c_i, and the class with maximum number of samples. Furthermore, RIR_i returns smaller values for majority classes while it returns larger values for minority classes. Since $n(t_k, c_i)$ is class-wide frequencies of t_k, $n(t_k, c_i)$ will probably be dominated by the majority classes. RIR_i; therefore, can strengthen minority classes if it is used in (2) (step 2 of AFE). Consequently, we propose to update (2) in AFE in favor of minority classes as formalized in (7)

$$w_{k,i} = \log\left(n(t_k, c_i)\right) \times IDF \times \log\left(1 + RIR_i^{1/p}\right), \tag{7}$$

where p is a parameter that can further scale RIR_i and can be set experimentally for the related data set. We call this method AFE-ME, *abstract feature extraction with minority elevation*, since weightings of minority classes are elevated. Introducing AFE-ME will help us address the class imbalance problem. Our intention to add such a parameter can be summarized as follows: *if the number of times a term occurs at least once in a minority class should be elevated against its number of occurrences against majority classes because majority classes have more documents for a term to appear*. The AFE method does not fulfill this assumption with its original version. To illustrate with an example, consider a case where a term, t_k, occurs 99 times in a minority class c_i, whose $RIR_i = 20$. The very same term, t_k, occurs 99 times in another class c_{i+1} whose RIR_{i+1} is 2, meaning that it is one of the majority classes compared to c_i. Assume that IDF for t_k is 2. Then, both weightings for t_k, which are $w_{k,i}$ and $w_{k,i+1}$, are equal, $\log(1 + 99) \times 2 = 4$, if AFE is utilized as is. However, we hypothesize that the weighting of t_k for c_i should be higher than c_{i+1} because the class-wide frequencies $(n(t_k, c_i))$ for both c_i and c_{i+1} are equal even though the number of documents appearing in c_{i+1} is 10 times greater than the number of documents appearing in c_i, which we can infer from RIR_i and RIR_{i+1}. Obviously, t_k is a more important term for c_i than c_{i+1}; therefore, it should be weighted accordingly. This is what we achieve by AFE-ME with RIR_i factor to elevate minority classes' weightings. If we implement the proposed weighting in AFE-ME with $p = 1$, $w_{k,i}$ becomes $4 \times \log\left(1 + 20^1\right) \approx 5.29$ while $w_{k,i+1}$ is $4 \times \log\left(1 + 2^1\right) \approx 1.91$.

4.2 Abstract feature extraction with relavant terms, AFE-RT

Recall that $w_{k,i}$ in (2), which calculates how much t_k affects the class, is the essence of AFE. AFE method achieves to get I number of features thanks to it. Recall that I is the number of classes. $w_{k,i}$ relies on $n(t_k, c_i)$, which is the class-wide frequency of t_k. However, AFE scales $n(t_k, c_i)$ only by IDF, which is an unsupervised method meaning that it does not consider the class membership of

t_k. Unlike TF that IDF is closely related, $n(t_k, c_i)$ is closely related to the class-membership information. Therefore, class membership information could be integrated into the equation. We propose to use an inter-class relevance indicator [19], A/B. Table 1 in Section 2 lists A and B. Recall that A is the number of documents belonging to c_i that t_k occurs at least once while B is the number of documents not belonging to c_i that t_k occurs at least once. The rationale behind our proposal is that *if t_k is a relevant term to $\boldsymbol{c_i}$, it should occur in more documents of c_i*. A/B ratio tends to be higher if t_k is highly relevant to c_i [19]. Therefore, we hypothesize that $w_{k,i}$ calculation in (2) of AFE should be substituted for (8), say AFE-RT, *abstract feature extraction with relevant terms*.

$$w_{k,i} = \log\left(n(t_k, c_i)\right) \times IDF \times \log\left(1 + \frac{A}{\max\left(1, B\right)}\right) \tag{8}$$

Besides AFE-RT expansion, we can combine it with our AFE-ME approach in the previous subsection to boost the accuracy for minority classes as well. So, our final equation for $w_{k,i}$ is given in (9), which we call AFE-MERT, *abstract feature extraction with minority elevation and relevant terms*.

$$w_{k,i} = \log\left(n(t_k, c_i)\right) \times IDF \times \log\left(1 + RIR_i^{1/p}\right) \times \log\left(1 + \frac{A}{\max\left(1, B\right)}\right) \tag{9}$$

4.3 Complexity analysis

Before giving the complexity analysis of the proposed AFE-MERT methods, Table 3 lists the abbreviations used in this subsection. Notice in the table that U and S are different. U is the total number of unique term in the corpus while S is the sum of the number of unique terms in each independent file and $S \geq U$.

Table 3 Notations used in the complexity analysis

Description	Notation
The number of documents	N
The number of terms a document, d_j, has	$\lvert d_j \rvert$
The number of total terms in corpus	$T = \sum_{j=1}^{N} \lvert d_j \rvert$
The number of unique terms in a document, d_j	$\lvert d_j \rvert_u$
The sum of all unique terms in each document	$S = \sum_{j=1}^{N} \lvert d_j \rvert_u$
The number of unique terms in corpus	U
The total number of classes	I

The first step, (1), requires each document in the corpus to be scanned to count class wide frequencies of each term, t_k. Therefore, this step requires $\mathcal{O}(T)$. Meanwhile, IDF calculation, which is in (2), needs the number of documents that each term occurs and it can be calculated at this step with no cost of extra repetition. After this partial computation, IDF can be finalized in $\mathcal{O}(U)$, by traversing all calculated values of all unique terms in the corpus. Similarly, A in Table 1, which is used in AFE-RT, is calculated here with no cost. In (2), the effect of each unique term to the relevant class, c_i, must be calculated. Then, the total number of repetitions would be $\mathcal{O}(U \times I)$. Our proposed method, AFE-MERT, is plugged into this part of the equation, (9). AFE-MERT is composed of AFE-ME and AFE-RT. The time complexity of the proposed AFE-ME method is $\mathcal{O}(I^2)$ because one round calculates max $(|C_1|, |C_2|, ..., |C_i|, ..., |C_{I-1}|, |C_I|)$ and the second round calculates RIR_i in Eq.6 for each class, c_i. AFE-RT requires both A and B in Table 1 to be calculated. Remember that A is calculated in step 1. The run time of B is $\mathcal{O}(U \times I)$ because it is calculated for each unique term and classes. After B is known, A/B ratio should be calculated for each unique term with global maximization function in $\mathcal{O}(U \times I)$. The final run time for AFE-RT; therefore, becomes $\mathcal{O}(2 \times U \times I)$. The total run time of the proposed method, AFE-MERT, in (9) including IDF calculation is $\mathcal{O}(U + I^2 + 3 \times U \times I)$. (3) calculates the overall weight of each document for all classes. We need to traverse the weights of unique terms in each document for each classes, $\mathcal{O}(S \times I)$. The last step is the normalization of the overall weights, which is $\mathcal{O}(N \times I)$. When all steps considered, the total complexity is $\mathcal{O}(T+U+I^2+3\times U\times I+S\times I+N\times I)$.

5 Experiments

5.1 Data sets

Three different data sets, Reuters-21578[1], 20_newsgroups[2], and ReutersUK, have been utilized throughout the experiments. Reuters-21578 is a well-known benchmark data set. We accessed the data set through Natural Language Toolkit[3], NLTK, a module in Python. The original Reuters-21578 is an imbalanced data set of 90 different classes with train and test sets. Some documents belong to more than one class; therefore, we pre-processed the data set. In our pre-processed version of Reuters-21578, we kept documents belonging to only one class from train set. There remain 58

[1]Reuters-21578 retrieved from Python Natural Language Toolkit

[2]20_newsgroups retrieved from http://qwone.com/~jason/20Newsgroups/

[3]https://www.nltk.org/

classes with 6,577 training documents. The IR for Reuters-21578 is 2,840 (the most dominant class has 2,840 training samples while the least one has just 1 training samples). Our second data set is 20_newsgroups data set (bydate version), which is a relatively balanced one. It includes 20 different classes. The data set includes 18846 documents with 11,314 train and 7,532 test documents. The IR is 1.591 (the most dominant class has 600 training samples while the least one has 377 training samples). Since we have only Reuters-21578 to represent the imbalanced class problem, we also collected data from ReutersUK[4] between November 13 and October 13, 2020. The collected data set includes 14 different classes and the IR for the data set is 123.47. The ReutersUK data set is available for researchers[5].

5.2 Evaluation criteria

Three different evaluation metrics have been used to evaluate the performance of the proposed methods. We use $f1\text{-}micro$, $f1\text{-}macro$, and MCC. Assuming true positives, false positives, and false negatives are given by TP, FP, and FN, respectively. $f1\text{-}micro$ and $f1\text{-}macro$, are given in (11) and (12) based on $f1\text{-}score$ in (10), respectively.

$$f1\text{-}score = \frac{2 \times precision \times recall}{precision + recall} \tag{10}$$

$$f1\text{-}micro = \frac{2\times\sum_{i=1}^{I} TP_{c_i}}{2\times\sum_{i=1}^{I} TP_{c_i}+\sum_{i=1}^{I} FP_{c_i}+\sum_{i=1}^{I} FN_{c_i}} \tag{11}$$

$$f1\text{-}macro = \frac{1}{I}\sum_{i=1}^{I} f1\text{-}score_{c_i} \tag{12}$$

$f1\text{-}macro$ takes averages of individual $f1\text{-}scores$ of all classes, so $f1\text{-}scores$ of minority and majority classes are weighted equally. This metric informs us whether minority classes are predicted correctly. On the other hand, how well minority classes are predicted cannot be observed by $f1\text{-}micro$ metric because $f1\text{-}micro$ calculates a global average. Since majority classes have many samples compared to minority classes, the $f1\text{-}micro$ score is dominated by the large sample size of majority classes. We use both metrics to demonstrate this obvious effect.

MCC can be utilized when classes are of different sizes [21]. It returns between -1 and 1. While 1 represents a perfect agreement between prediction and observations, -1 is opposite. Unlike metrics that are influenced by the presence of majority classes like $f1\text{-}micro$, MCC is more informative to express the relation between predictions and observations. The formulae for MCC is given in (13).

$$MCC = \frac{TP\times TN - FP\times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \tag{13}$$

5.3 Methodology

Although Rueters-21578 and 20_newsgroups data sets have already both training and test samples, we use 10-fold cross validation for all data sets in our experiments. Therefore, we only utilize training samples of Rueters-21578 and 20_newsgroups data sets. Since some classes have less than 10 samples and 10-fold cross validation is conducted, we did random oversampling to have at least 10 samples.

We applied some pre-processing before the feature extraction. We tokenized each word, removed punctuation, and kept the words whose length is greater than 1. Then, each word is lowered, stop words are removed. Finally, we did the stemming with Porter stemming algorithm [23] via NLTK.

5.4 Classifiers

The classifiers used in the experiments are briefly given in this subsection. We utilize KNN, NC, and SVM, which are frequently used classifiers in text classification literature [5, 8, 9, 17, 29].

Unlike eager learners that construct a model when training samples are given, KNN algorithm is a lazy learner, which waits until a test sample to be given to make a prediction [14]. Given a test sample, KNN algorithm computes its similarity to the training samples. The closest k training samples are determined as neighbors, where *k-nearest neighbor* analogy derives from. By majority voting, the test sample is assigned to the most frequent class among selected *k-nearest neighbors*. In KNN, the number of neighbors, k, is an important parameter to report for reproducibility. k is 10 in our experiments.

In NC, a centroid is calculated from training samples for each class. A centroid defines the center of class samples. It can be calculated based on statistics such as mean or median. Each class is represented by its centroid, and the model creation is to discover centroids for each class. Since each class is represented by a prototype, centroids in this case, an unseen sample is assigned to the class whose centroid is the closest.

SVM classifies both linear and non-linear data and is a suitable method to work with high dimensional data, which is the case for text classification. SVM represents a decision boundary to separate samples of one class from another by using samples from the training set, which are known as support vectors [27]. It transforms the original data into a

[4] http://uk.retuers.com

[5] https://github.com/mrtokk

higher dimension, and this dimension is used to search a decision boundary with maximal margin [27]. Each decision boundary is associated with a pair of hyper-plane. A hyper-plane is parallel to the decision boundary and it touches data points (samples), which are indeed support vectors. The distance from the decision boundary to the support vectors is called margin, and SVM looks for maximal margin hyper-plane. Larger margins tend to generalize future samples better than the smaller margins [27]. In the experiments, we used linear SVM through *scikit-learn*[6] module's LinearSVC [12] class with default settings in Python.

5.5 Experimental results

In this section, we experimentally test our proposed improvements on AFE. The very first two experiments independently scrutinize the effects of the introduced methods, AFE-ME and AFE-RT, respectively. We aim to show their independent contributions to the original AFE. In line with this idea, in the first experiment, we determine the optimal value for the parameter p in our AFE-ME method and discuss the effect of AFE-ME, especially on the $f1\text{-}macro$ score when used as a classification method. In the second experiment, AFE-RT method is compared with AFE if they are used as classifiers. By doing so, we aim to represent the contribution made solely by AFE-RT method. After determining the optimal p value and discussing the contribution made by AFE-ME and AFE-RT individually, we demonstrate the improvement by our AFE-MERT approach compared to the original AFE considering the evaluation metrics ($f1\text{-}micro$, $f1\text{-}macro$, and MCC) if they are used as classifiers. Then, we utilize AFE-MERT and AFE as feature extraction and dimension reduction techniques to be used in well-known classifiers and compare them. In this experiment, we also include AFE-MERT and AFE as a classifier to compare them with well-known classifiers. Last, we utilize the state-of-the-art term weighting schemes to compare them with AFE-MERT. This experiment shows how the proposed method performs against well-known term weighting schemes in the literature.

5.5.1 Experiment I: Determining *p* parameter and effect of AFE-ME on *f*1-*macro* score

Recall that the $f1\text{-}macro$ score helps us understand how well the samples from minority classes are predicted. Since the primary concentration in this study is the class imbalance problem, demonstrating that our proposed AFE-ME method, which plugs RIR into AFE, improves the $f1\text{-}macro$ scores is crucial. In this experiment, we report all of the three metrics but particularly focus on the $f1\text{-}macro$ score to pick the optimal p.

We hypothesize that introducing RIR factor into AFE by the proposed AFE-ME method will result in an increase in terms of $f1\text{-}macro$ score for imbalanced data sets, which are Ruters-21578 and ReutersUK in our case. However, the degree of imbalance in 20_newsgroups data set is moderate; therefore,we do not expect to see dramatic changes in the results when AFE-ME method is applied for 20_newsgroups data set. The experimental results are given in Fig. 1.

As one can see from the figure, the proposed method with different p values always performs considerably better than the original AFE method in terms of $f1\text{-}macro$ metric for the imbalanced data sets. This is true for MCC and $f1\text{-}micro$ results as well for all p values except $p = 1$. Although AFE-ME outperforms the original AFE method in terms of $f1\text{-}macro$ when p is 1, it falls behind in terms of $f1\text{-}micro$ and MCC for Reuters-21578 and ReutersUK data sets. In general, one can conclude from the results that the proposed method, AFE-ME, outperforms the original AFE method significantly when the data set is imbalanced. Besides, in line with our expectation, the proposed method does not introduce substantial improvement when the data set is moderately imbalanced. One can confirm from the figure that $f1\text{-}macro$, $f1\text{-}micro$, and MCC are around AFE when our proposed method is applied with different p values.

We pick 2 as an optimal p value since it is the one that contributes the most to the $f1\text{-}macro$ score, which is the main intention to conduct this experiment. For Reuters-21578 data set, the $f1\text{-}macro$ result is around 0.149 when AFE is used. The proposed method, AFE-ME, outperforms it with a large margin by taking it around 0.690. The improvement achieved by AFE-ME is around 363%. A similar observation can be made for ReutersUK data set, where $f1\text{-}macro$ is around 0.199 with AFE while the proposed method, AFE-ME, improves it by around 190% and its $f1\text{-}macro$ is around 0.578. This experiment clearly shows that introducing RIR as a factor in our AFE-ME method significantly improves the $f1\text{-}macro$ results for imbalanced data sets. Thus, unless otherwise stated, the following experiments use $p = 2$.

5.5.2 Experiment II: The effect of AFE-RT on the general accuracy

The previous experiment scrutinizes the effects of AFE-ME on the prediction of samples from minority classes. In this experiment, we would like to discuss the effects of AFE-RT when it is used as a classification method. The idea in AFE-RT is to scale the terms that are relevant to the class. Therefore, the hypothesis is that AFE-RT could achieve an

[6] scikit-learn: https://scikit-learn.org/

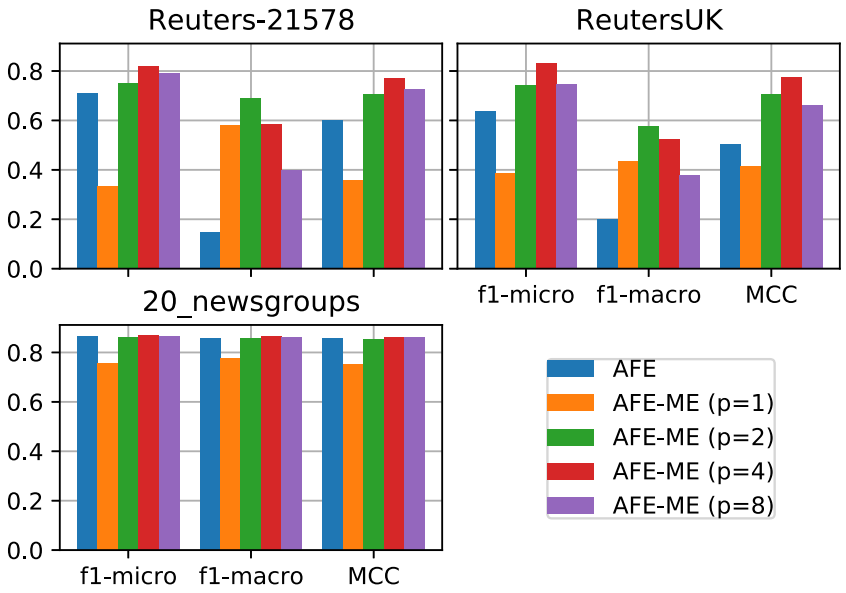


Fig. 1 AFE and AFE-ME used as a classifier

increase in the evaluation metrics regardless of the data set type, imbalanced or moderately imbalanced. Note that the maximum globalization function is used to get a unique value for each term and class.

The experimental results are given Fig. 2. The proposed AFE-RT method outdoes the original AFE method for all data sets and metrics, $f1\text{-}micro$, $f1\text{-}macro$ or MCC. Notice that although AFE-ME reports a moderate change in balanced data set, 20_newsgroups, in the previous experiment, which we expected so, AFE-RT is now outperforming. This experiment displays that promoting relevant terms has positive effects when the method is used as a classifier.

5.5.3 Experiment III: Comparing AFE-MERT and AFE as a classifier

The previous two experiments show the independent contributions of our proposed methods. In this experiment, we discuss the effects of AFE-MERT, which is a combined version of AFE-ME and AFE-RT. Our hypothesis based on findings from Experiment I and II is that the AFE-MERT method will considerably outperform AFE in all metrics. Experimental results are reported in Fig. 3.

It can be verified from the figure that the proposed AFE-MERT method performs significantly better than the original AFE method when used as a classifier. For imbalanced data sets, Reuters-21578 and ReutersUK, $f1\text{-}micro$, $f1\text{-}macro$, and MCC are considerably higher. Although the intention is to improve $f1\text{-}macro$ results to support samples from minority classes in imbalanced data sets, the $f1\text{-}micro$ and MCC results are also worth mentioning. $f1\text{-}micro$ is around 0.828 and 0.854 for Ruters-21578 and ReutersUK data sets, respectively, when AFE-MERT is applied. This means that the original AFE method is improved around 16% and 34%, respectively (corresponding values are around 0.711 and 0.638, respectively). Similar observations can be made for MCC results as well. In terms of $f1\text{-}macro$ results, the improvement made by AFE-MERT method is phenomenal with the imbalanced data sets. AFE-MERT outputs around 0.659 and 0.630 for Ruters-21578 and ReutersUK data sets, respectively, while AFE outputs around 0.149 and 0.199 for the same data sets. When the balanced data set, 20_newsgroups, is considered, the AFE-MERT method again outdoes the original AFE in all evaluation criteria. This verifies that our proposed method, AFE-MERT, can replace AFE for both imbalanced and moderately imbalanced data sets when used as a classifier. The experimental results are also important to express that the proposed AFE-MERT method becomes a competitive classifier that will be discussed in the next experiment.

5.5.4 Experiment IV: AFE-MERT and AFE as feature extraction and dimension reduction method to be used in classifiers

After comparing the proposed methods and AFE in many ways, the idea in this experiment is to use AFE-MERT and AFE as feature extraction and dimension reduction method in well-known classifiers. Besides, we compare the results from these classifiers with the original AFE and the proposed AFE-MERT method as classifiers to show that AFE-MERT is now a competitive classifier. The related results are given in Table 4.

When Reuters-21578 data set is considered, the proposed AFE-MERT method outperforms the original AFE method as a feature extraction and dimension reduction tool for

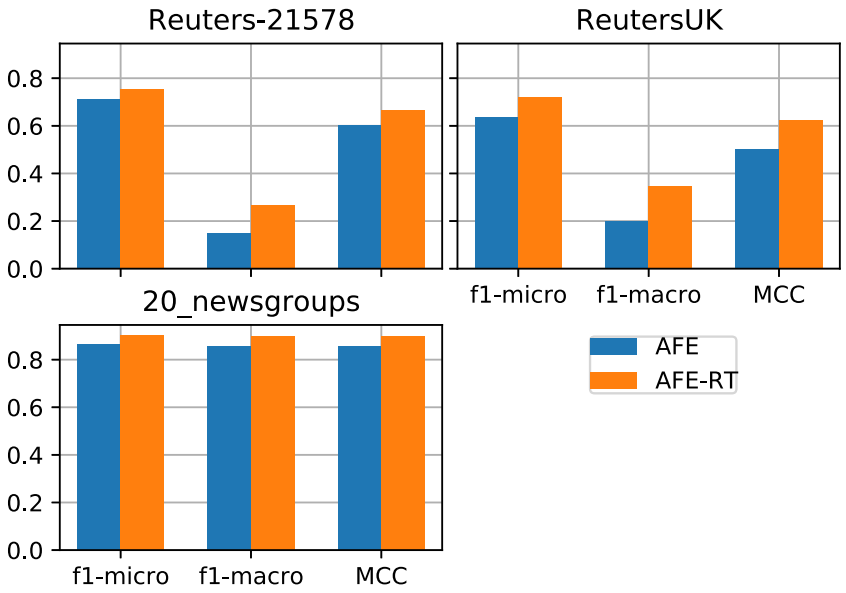


Fig. 2 AFE and AFE-RT used as a classifier

the corresponding classifiers in all metrics. Although the AFE method falls dramatically behind against KNN, NC, and SVM as a classifier, AFE-MERT outperforms NC and SVM in terms of $f1\text{-}micro$, KNN and SVM in terms of $f1\text{-}macro$, and NC and SVM in terms of MCC. In Reuters-UK data set, KNN classifier records similar results regardless of either AFE-MERT or AFE is utilized as a feature extraction and dimension reduction method while AFE-MERT outperforms with SVM classifier. Among all experimental settings in this experiment, AFE is only better than AFE-MERT when NC is utilzed for ReutersUK data set. Similar to the previous experiment, AFE-MERT is a competitive classifier while AFE as a classifier falls dramatically behind compared to KNN, NC, and SVM. For 20_newsgroups data set, corresponding results for KNN, NC, and SVM in all metrics show that the proposed AFE-MERT method outperforms the original AFE.

To sum up, this experiment demonstrates that the proposed AFE-MERT method is a better replacement for AFE as a feature extraction and dimension reduction. In previous experiments, we have already discussed that AFE-MERT outdoes AFE as a classifier as well. Therefore, we can conclude that introducing imbalance ratio- and relevant term-based scales into AFE not only helps predict minority classes but also improve the general accuracy.

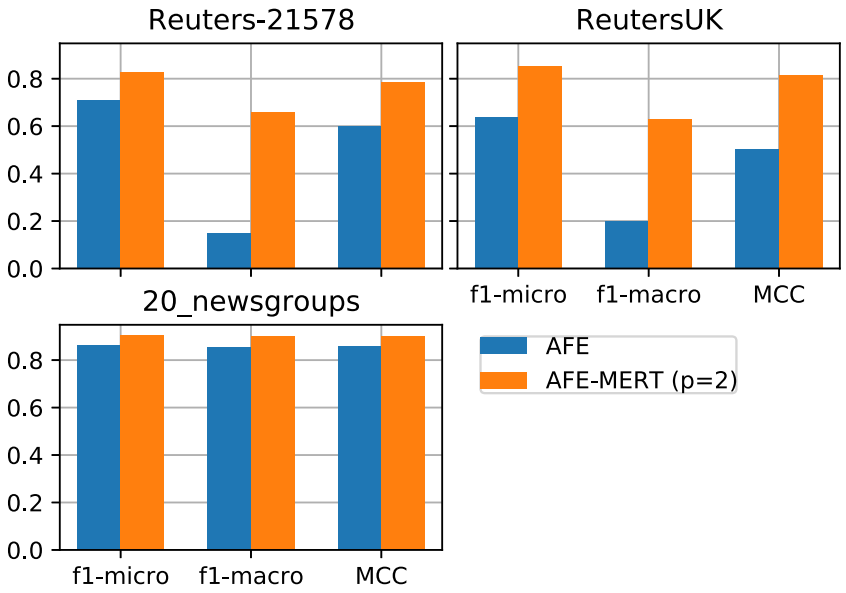


Fig. 3 AFE-MERT and AFE used as a classifier

Table 4 AFE-MERT and AFE when used as feature extraction and dimension reduction method for well-known classifiers

Data set	Feature extraction & dimension reduction method	Classifier	$f1\text{-}micro$	$f1\text{-}macro$	MCC
Reuters-21578		AFE	0.711	0.150	0.601
	AFE	KNN	0.880	0.449	0.839
		NC	0.793	0.680	0.745
		SVM	0.768	0.133	0.680
	AFE-MERT	AFE-MERT	0.828	0.659	0.784
		KNN	0.885	0.487	0.845
		NC	0.806	0.691	0.760
		SVM	0.806	0.212	0.735
ReutersUK	AFE	AFE	0.638	0.199	0.503
		KNN	0.865	0.674	0.819
		NC	0.845	0.709	0.800
		SVM	0.796	0.408	0.723
	AFE-MERT	AFE-MERT	0.854	0.630	0.813
		KNN	0.866	0.672	0.820
		NC	0.825	0.700	0.772
		SVM	0.833	0.528	0.777
20_newsgroups	AFE	AFE	0.863	0.855	0.857
		KNN	0.913	0.911	0.908
		NC	0.869	0.868	0.862
		SVM	0.898	0.890	0.893
	AFE-MERT	AFE-MERT	0.904	0.900	0.899
		KNN	0.920	0.918	0.916
		NC	0.885	0.886	0.879
		SVM	0.915	0.912	0.911

5.5.5 Experiment V: Comparison with state-of-the-art term weighting and dimension reduction approaches

The previous experiments discuss the superiority of our method, AFE-MERT, to the original method, AFE, by several experimental settings. This experiment, on the other hand, compares AFE and AFE-MERT with the state-of-the-art term weighting schemes, TF-IDF, TF-IDF-ICF [24], TF-IDF-ICSDF [24], TF-RF [17], TF-PROB [19], TF-IGM [5], and TF-MONO [10], and dimension reduction techniques [6], chi-square, odds ratio, and information gain, in text classification. Recall that the original AFE method, which we improved, is not only a supervised term weighting scheme (we usually call this property of AFE as feature extraction and dimension reduction method in the paper) but also a classifier. Since the term weighting schemes in the literature do not have the property of dimension reduction; thus, we will utilize feature selection methods, chi-square, odds ratio, and information gain, to reduce the number of features (terms in our case) that defines documents. Remember that the original method of AFE reduces the number of dimensions into the number of available classes; therefore, we select the most informative 58, 14, and 20 features with above-mentioned feature selection methods for Reuters-21578, ReutersUK, and 20_newsgroups data sets, respectively. Note that we utilize the global maximization function and notice that these number of features are fairly low. For reproducibility, note that α is 6 as [5] and [10] did in their papers. The experimental results are reported in Tables 5, 6, and 7 for each data set. We only report $f1\text{-}macro$ results since our primary consideration is class imbalance problem and reporting $f1\text{-}micro$ and MCC would make the table hard to follow. The tables include both AFE and AFE-MERT, and notice that the reported results for AFE and AFE-MERT are not produced by chi-square, odds ratio, or information gain feature selection methods. They already include feature selection property; therefore, the reported results are same for each feature selection method for comparison purposes. Before discussing experimental findings, we should give AFE credit for its superiority to the other dimension reduction methods as discussed in [3]. It is already overachieving if the number of selected features is equal to the number of classes.

At first glance, it is obvious that both AFE and our method, AFE-MERT, outperform the term weighting

Table 5 Comparison with the state-of-the-art term weighting schemes when the number of features are equal to the number of classes, Reuters-21578 data set

Term weighting	Chi-square			Odds ratio			Information gain		
	KNN	NC	SVM	KNN	NC	SVM	KNN	NC	SVM
TF-IDF	0.298	0.340	0.350	0.274	0.198	0.189	0.449	0.387	0.391
TF-IDF-ICF	0.300	0.341	0.350	0.276	0.191	0.200	0.463	0.391	0.365
TF-IDF-ICSDF	0.273	0.321	0.336	0.264	0.188	0.186	0.420	0.343	0.379
TF-RF	0.297	0.336	0.351	0.259	0.199	0.173	0.435	0.340	0.354
TF-PROB	0.296	0.346	0.348	0.276	0.219	0.191	0.455	0.293	0.330
TF-IGM	0.298	0.336	0.346	0.290	0.220	0.190	0.452	0.377	0.405
TF-MONO	0.295	0.337	0.350	0.262	0.212	0.179	0.440	0.386	0.384
AFE	0.449	0.680	0.133	0.449	0.680	0.133	0.449	0.680	0.133
AFE-MERT	0.487	0.691	0.212	0.487	0.691	0.212	0.487	0.691	0.212

Table 6 Comparison with the state-of-the-art term weighting schemes when the number of features are equal to the number of classes, ReutersUK data set

Term weighting	Chi-square			Odds ratio			Information gain		
	KNN	NC	SVM	KNN	NC	SVM	KNN	NC	SVM
TF-IDF	0.205	0.205	0.254	0.151	0.152	0.188	0.263	0.326	0.314
TF-IDF-ICF	0.206	0.207	0.254	0.151	0.152	0.188	0.265	0.328	0.313
TF-IDF-ICSDF	0.209	0.202	0.259	0.152	0.146	0.188	0.259	0.324	0.310
TF-RF	0.209	0.206	0.258	0.152	0.145	0.188	0.262	0.317	0.315
TF-PROB	0.208	0.210	0.257	0.149	0.146	0.188	0.258	0.321	0.313
TF-IGM	0.207	0.208	0.254	0.151	0.152	0.188	0.263	0.316	0.311
TF-MONO	0.208	0.204	0.258	0.150	0.152	0.188	0.264	0.319	0.314
AFE	0.674	0.709	0.408	0.674	0.709	0.408	0.674	0.709	0.408
AFE-MERT	0.672	0.700	0.528	0.672	0.700	0.528	0.672	0.700	0.528

Table 7 Comparison with the state-of-the-art term weighting schemes when the number of features are equal to the number of classes, 20_newsgroups data set

Term weighting	Chi-square			Odds ratio			Information gain		
	KNN	NC	SVM	KNN	NC	SVM	KNN	NC	SVM
TF-IDF	0.233	0.357	0.362	0.097	0.097	0.098	0.249	0.391	0.367
TF-IDF-ICF	0.233	0.362	0.362	0.097	0.097	0.098	0.251	0.398	0.368
TF-IDF-ICSDF	0.229	0.345	0.352	0.097	0.097	0.098	0.244	0.334	0.342
TF-RF	0.233	0.355	0.362	0.097	0.097	0.098	0.246	0.376	0.367
TF-PROB	0.233	0.362	0.361	0.097	0.097	0.098	0.250	0.395	0.368
TF-IGM	0.234	0.362	0.361	0.097	0.097	0.098	0.250	0.397	0.367
TF-MONO	0.232	0.355	0.362	0.097	0.097	0.098	0.247	0.376	0.367
AFE	0.911	0.868	0.889	0.911	0.868	0.889	0.911	0.868	0.889
AFE-MERT	0.918	0.886	0.912	0.918	0.886	0.912	0.918	0.886	0.912

schemes in the literature when the number of features is reduced to the number of classes by any of the feature selection method. The only exception is the Reuters-21578 data set with SVM classifier of chi-square and information gain feature selection. This is actually the effect of the feature reduction technique because the difference is more dramatic when the number of features is 14 and 20 for ReutersUK and 20_newsgroups. In these two data sets, regardless of the classifier, both AFE and AFE-MERT achieve strikingly better results. In fact, this can be attributed to the AFE method's ability to extract more informative features [3]. Between AFE and AFE-MERT, as previously discussed in earlier experiments, our proposed method, AFE-MERT, is usually better. In Reuters-21578 data set, both AFE and AFE-MERT again outperform the rest except SVM classifier. However, while utilizing Reuters-21578 data set, the gap is narrower than it is with ReutersUK and 20_newsgroups. The gap narrows because the Reuters data set has 58 classes; therefore, 58 features are selected by chi-square, odds ratio, and information gain which is greater than 14 and 20 used in ReutersUK and 20_newsgroups. This experiment up to now displays that both AFE and AFE-MERT achieve spectacularly better results than the state-of-the-art term weighting methods if the associated features are reduced to the number of classes as AFE does. For a different view, we would like to increase the number of features to 500 for the rest of the term weighting schemes to see how these schemes perform. We give a comparison in Tables 8, 9, and 10. Note that the number features that AFE and AFE-MERT select is still equal to the number of classes.

When the number of features is increased up to 500 for the term weighting schemes, we can argue that our proposed scheme, AFE-MERT, is competitive although the number of features is relatively low compared to 500. Reuters-21578 is particularly important because it is the most imbalanced data set in our experiments. With NC classifier, AFE-MERT is the best performing method when the feature selection is odds ratio. AFE-MERT is better than the most of the term weighting schemes in terms of $f1\text{-}macro$ when the feature selection is either chi-square or information gain. With KNN classifier, AFE-MERT can only achieve to outperform when the feature selection is odds ratio. With SVM classifier, AFE-MERT falls behind. With ReutersUK data set, both AFE and AFE-MERT are the best when the classifier is KNN. Only TF-IDF and TF-IDF-ICF achieve to outperform when the feature selection method is information gain. In terms of NC, AFE-MERT is the best performing one when the feature selection is odds ratio. When the feature selection is either chi-square or information gain, TF-IDF and TF-IDF-ICF are outperforming. With SVM classifier, AFE-MERT is the best performing one with odds ratio feature selection method while it does not introduce an improvement with chi-square and information gain. With 20_newsgroup data set, there is a clear domination of AFE-MERT against other term weighting schemes. In all cases, AFE-MERT, which relies on 20 features only, is the best performing scheme by large margin. This experiment briefly shows that AFE-MERT is better than odds ratio in terms of $f1\text{-}macro$ as dimension reduction tool although its number of features are relatively low compared to 500. If other feature selection methods are preferred to reduce the number features, the choice of classifier might be crucial.

To summarize, this experiment performs two types of comparison of our proposed method with the term weighting schemes in the literature. The first part of the experiment reduces the number of features down to the number of classes as AFE and AFE-MERT do. The second part of the experiment shows that even the number of features is increased to a large value, 500 in our case, the experimental results suggest that our proposed method achieves to compete with the state-of-the-art term weighting schemes and feature selection methods.

5.5.6 Statistical analysis

We also apply t-test to show whether the improvements achieved by our proposed methods are statistically significant. Since we are primarily interested in imbalanced text classification, the statistics are calculated using $f1\text{-}macro$ in this subsection.

Table 11 gives the corresponding p-values (one-tailed paired t-test) for the improvements our methods achieve against AFE. This table corresponds to the first four experiments, Sections 5.5.1, 5.5.2, 5.5.3, and 5.5.4. Notice that we call t-test's p as p-value not to confuse it with our methods' parameter p. The improvements achieved by the proposed methods, AFE-ME, AFE-RT, and their combination AFE-MERT, are always statistically significant (very low p-values) when used as classifiers in the first three experiments. When the proposed method used as feature extraction and dimension reduction technique to be used in classifiers in Section 5.5.4, the improvements provided by the proposed method, AFE-MERT, are statistically significant as well. The least confidence score in this table is above 90% confidence level.

The last experiment we perform compares the proposed method and the state-of-the-art term weighting schemes and dimension reduction (feature selection) techniques in Section 5.5.5. AFE and AFE-MERT are compared with seven different term weighting schemes and three different dimension reduction technique using three different classifiers, KNN, NC, and SVM. Therefore, AFE-MERT takes part in 63 different comparisons in each data set. Since we have three different data sets, the number of comparisons

Table 8 Comparison with the state-of-the-art term weighting schemes when the number of features are 500, Reuters-21578 data set

Term weighting	Chi-square			Odds ratio			Information gain		
	KNN	NC	SVM	KNN	NC	SVM	KNN	NC	SVM
TF-IDF	0.658	0.713	0.788	0.497	0.421	0.476	0.663	0.756	0.849
TF-IDF-ICF	0.621	0.711	0.767	0.511	0.415	0.467	0.674	0.724	0.848
TF-IDF-ICSDF	0.565	0.605	0.742	0.433	0.368	0.451	0.530	0.651	0.814
TF-RF	0.589	0.581	0.718	0.417	0.318	0.365	0.571	0.589	0.782
TF-PROB	0.593	0.578	0.665	0.473	0.371	0.421	0.579	0.548	0.687
TF-IGM	0.609	0.685	0.792	0.459	0.386	0.443	0.604	0.693	0.815
TF-MONO	0.609	0.637	0.737	0.426	0.334	0.345	0.603	0.635	0.819
AFE	0.449	0.680	0.133	0.449	0.680	0.133	0.449	0.680	0.133
AFE-MERT	0.487	0.691	0.212	0.487	0.691	0.212	0.487	0.691	0.212

Table 9 Comparison with the state-of-the-art term weighting schemes when the number of features are 500, ReutersUK data set

Term weighting	Chi-square			Odds ratio			Information gain		
	KNN	NC	SVM	KNN	NC	SVM	KNN	NC	SVM
TF-IDF	0.667	0.739	0.719	0.330	0.322	0.414	0.681	0.717	0.701
TF-IDF-ICF	0.660	0.732	0.715	0.331	0.327	0.414	0.692	0.721	0.713
TF-IDF-ICSDF	0.588	0.655	0.708	0.314	0.317	0.412	0.562	0.625	0.648
TF-RF	0.596	0.627	0.723	0.318	0.271	0.395	0.553	0.577	0.673
TF-PROB	0.558	0.606	0.664	0.332	0.323	0.410	0.541	0.568	0.650
TF-IGM	0.662	0.728	0.722	0.332	0.328	0.414	0.624	0.688	0.702
TF-MONO	0.619	0.671	0.715	0.331	0.304	0.406	0.594	0.643	0.688
AFE	0.674	0.709	0.408	0.674	0.709	0.408	0.674	0.709	0.408
AFE-MERT	0.672	0.700	0.528	0.672	0.700	0.528	0.672	0.700	0.528

Table 10 Comparison with the state-of-the-art term weighting schemes when the number of features are 500, 20_newsgroups data set

Term weighting	Chi-square			Odds ratio			Information gain		
	KNN	NC	SVM	KNN	NC	SVM	KNN	NC	SVM
TF-IDF	0.707	0.705	0.769	0.387	0.402	0.407	0.701	0.712	0.777
TF-IDF-ICF	0.724	0.737	0.767	0.387	0.402	0.407	0.720	0.740	0.776
TF-IDF-ICSDF	0.694	0.664	0.758	0.387	0.406	0.407	0.635	0.612	0.761
TF-RF	0.670	0.592	0.767	0.387	0.400	0.408	0.611	0.555	0.773
TF-PROB	0.718	0.661	0.763	0.387	0.395	0.408	0.708	0.659	0.764
TF-IGM	0.730	0.716	0.772	0.387	0.401	0.408	0.729	0.718	0.779
TF-MONO	0.666	0.596	0.761	0.387	0.397	0.408	0.627	0.574	0.771
AFE	0.911	0.868	0.889	0.911	0.868	0.889	0.911	0.868	0.889
AFE-MERT	0.918	0.886	0.912	0.918	0.886	0.912	0.918	0.886	0.912

Table 11 Statistical significance of the improvements achieved by the proposed methods when used as a classifier

	Data set	p-value	Reference
AFE vs AFE-ME (p=1)	Reuters-21578	6.30-e12	Section 5.5.1
AFE vs AFE-ME (p=1)	ReutersUK	2.37-e7	
AFE vs AFE-ME (p=2)	Reuters-21578	5.47-e13	
AFE vs AFE-ME (p=2)	ReutersUK	2.42e-9	
AFE vs AFE-ME (p=2)	20_newsgroups	0.073	
AFE vs AFE-ME (p=4)	Reuters-21578	1.43-e13	
AFE vs AFE-ME (p=4)	ReutersUK	1.43e-11	
AFE vs AFE-ME (p=4)	20_newsgroups	3.55e-4	
AFE vs AFE-ME (p=8)	Reuters-21578	4.26-e10	
AFE vs AFE-ME (p=8)	ReutersUK	2.78e-9	
AFE vs AFE-ME (p=8)	20_newsgroups	9.30e-5	
AFE vs AFE-RT	Reuters-21578	4.89-e9	Section 5.5.2
AFE vs AFE-RT	ReutersUK	9.05e-9	
AFE vs AFE-RT	20_newsgroups	2.69e-9	
AFE vs AFE-MERT (p=2)	Reuters-21578	8.70-e13	Section 5.5.3
AFE vs AFE-MERT (p=2)	ReutersUK	4.83e-10	
AFE vs AFE-MERT (p=2)	20_newsgroups	3.84e-10	
AFE vs AFE-MERT KNN	Reuters-21578	0.001	Section 5.5.4
AFE vs AFE-MERT KNN	20_newsgroups	7.51e-5	
AFE vs AFE-MERT NC	Reuters-21578	0.080	
AFE vs AFE-MERT KNN	20_newsgroups	9.76e-6	
AFE vs AFE-MERT SVM	Reuters-21578	8.42e-8	
AFE vs AFE-MERT SVM	ReutersUK	7.50e-9	
AFE vs AFE-MERT SVM	20_newsgroups	2.26e-8	

becomes 189. We do not give statistical table due to the large number of comparison. Instead, we will verbalize the results.

When the number of features are reduced to the number of classes, out of 183 possible comparisons in corresponding Tables 5, 6, and 7, the number of comparisons that the proposed method outperforms is 175. The results achieved by the proposed method are always statistically significant except one case (the data set is Reuters, feature selection is odds ratio, classifier is SVM, $p\text{-}value = 0.055 > 0.05$). When the number of features are 500 for term weighting schemes, our method outperforms in 126 cases out of 189. Among these 126 cases, where AFE-MERT is better, 120 of them are statistically significant.

6 Conclusion

The class imbalance occurs when some classes have many training samples while some others have few. The class imbalance can be a burden to fit accurate classifiers for minority classes due to the fact that the model could bias toward majority classes. Like many problems in machine learning, the class imbalance problem is handled in the text classification literature as well. The purpose of this study is to tackle class imbalance problem in abstract feature extraction [3], AFE, method. AFE is a powerful tool in text classification that can serve as a feature extraction, dimension reduction method and classifier. We scrutinize abstract feature extraction in detail and propose to plug *relative imbalance ratio* based factor into it to elevate minority class representations, which is called abstract feature extraction with minority elevation, AFE-ME. Empirical results show that the proposed method clearly improves AFE method when used as a classifier for very imbalanced data sets. We also propose to integrate a *relevant term* factor, called abstract feature extraction with relevant terms, AFE-RT. The proposed method of integrating relevant terms outperforms the original AFE method as a classifier in terms of all evaluation metrics, $f1\text{-}macro$, $f1\text{-}micro$, and *Matthew's correlation coefficient*. Finally, we achieve remarkably better results

when these two approaches are combined together, AFE-MERT. When our proposed methods used as a feature extraction and dimension reduction method to be used in well-known classifiers, it outperforms abstract feature extraction method. Moreover, the experimental results show that the proposed method becomes a competitive classifier when compared with the well-known classifiers. Last, we experimentally compare our proposed method, AFE-MERT, with the state-of-the-art term weighting schemes and feature selection techniques, our method achieves decent results when the number of selected features in term weighting scheme is limited to the number of classes.

This study confirms that the proposed method to strengthen minority classes with relative imbalance ratio achieves decent results. Based on promising results that relative imbalance ratio achieved on the class imbalance problem, as a future work, we plan to investigate whether this technique can be applied to other term weighting schemes. Although there are some studies in the term weighting literature focusing on class imbalance problem, more attention is needed. Moreover, machine learning literature has a deep literature of multi-party computation that allows multi parties work together on a joint problem; however, we observe that text classification or term weighting literature lacks such insights. Some parties could come together to increase their accuracy scores in text classification. Such collaborations could help overcome the imbalanced text classification problem as well. To achieve such goals, protocols that govern multi-party computation and communication and related privacy or security mechanisms for participating parties must be also studied. We; therefore, plan to investigate protocols that can help overcome the class imbalance problem by collaboration.

Declarations

Conflict of Interests The authors declare that they have no conflict of interest.

References

1. Aggarwal CC, Zhai C (2012) A survey of text classification algorithms. In: Mining text data, Springer, Boston, pp 163–222
2. Alsaeedi A (2020) A survey of term weighting schemes for text classification. Int J Data Min Model Manag 12(2):237
3. Biricik G, Diri B, Sonmez AC (2012) Abstract feature extraction for text classification. Turk J Electr Eng Comput Sci 20(Sup. 1):1137–1159
4. Buda M, Maki A, Mazurowski MA (2018) A systematic study of the class imbalance problem in convolutional neural networks. Neural Netw 106:249–259
5. Chen K, Zhang Z, Long J, Zhang H (2016) Turning from TF-IDF to TF-IGM for term weighting in text classification. Expert Syst Appl 66:245–260
6. Debole F, Sebastiani F (2004) Supervised term weighting for automated text categorization. In: Text mining and its applications, Springer, Berlin, pp 81–97
7. Deisy C, Gowri M, Baskar S, Kalaiarasi SMA, Ramraj N (2010) A novel term weighting scheme midf for text categorization. J Eng Sci Technol 5(1):94–107
8. Dogan T, Uysal AK (2019a) Improved inverse gravity moment term weighting for text classification. Expert Syst Appl 130:45–59
9. Dogan T, Uysal AK (2019b) On term frequency factor in supervised term weighting schemes for text classification. Arab J Sci Eng 44(11):9545–9560
10. Dogan T, Uysal AK (2020) A novel term weighting scheme for text classification: Tf-mono. J Inf 14(4):101076. https://doi.org/10.1016/j.joi.2020.101076, https://www.sciencedirect.com/science/article/pii/S1751157720300705
11. Domo (2021) Domo resource - data never sleeps 8.0. https://www.domo.com/learn/data-never-sleeps-8
12. Fan RE, Chang KW, Hsieh CJ, Wang XR, Lin CJ (2008) LIBLINEAR: A library for large linear classification. J Mach Learn Res 9:1871–1874
13. Haixiang G, Yijing L, Shang J, Mingyun G, Yuanyue H, Bing G (2017) Learning from class-imbalanced data: Review of methods and applications. Expert Syst Appl 73:220–239
14. Han J, Kamber M, Pei J (2012) Data mining: Concepts and techniques. Morgan Kaufmann, Oxford
15. He H, Garcia EA (2009) Learning from imbalanced data. IEEE Trans Knowl Data Eng 21(9):1263–1284
16. Kowsari MJ, Heidarysafa M, Barnes B (2019) Text classification algorithms: a survey. Inf (Basel) 10(4):150
17. Lan M, Tan CL, Su J, Lu Y (2009) Supervised and traditional term weighting methods for automatic text categorization. IEEE Trans Pattern Anal Mach Intell 31(4):721–735
18. Li Y, Sun G, Zhu Y (2010) Data imbalance problem in text classification. In: 2010 Third International Symposium on Information Processing. IEEE
19. Liu Y, Loh HT, Sun A (2009) Imbalanced text classification: a term weighting approach. Expert Syst Appl 36(1):690–701
20. Loyola-González O, Martínez-Trinidad JF, Carrasco-Ochoa JA, García-Borroto M (2016) Study of the impact of resampling methods for contrast pattern based classifiers in imbalanced databases. Neurocomputing 175:935–947
21. Matthews BW (1975) Comparison of the predicted and observed secondary structure of T4 phage lysozyme. Biochim Biophys Acta 405(2):442–451
22. Ortigosa-Hernández J, Inza I, Lozano JA (2017) Measuring the class-imbalance extent of multi-class problems. Pattern Recognit Lett 98:32–38
23. Porter MF (2006) An algorithm for suffix stripping. Program 40(3):211–218
24. Ren F, Sohrab MG (2013) Class-indexing-based term weighting for automatic text classification. Inf Sci 236:109–125. https://doi.org/10.1016/j.ins.2013.02.029, https://www.sciencedirect.com/science/article/pii/S0020025513001461
25. Sabbah T, Selamat A, Selamat MH, Al-Anzi FS, Viedma EH, Krejcar O, Fujita H (2017) Modified frequency-based term weighting schemes for text classification. Appl Soft Comput 58:193–206
26. Sun Y, Kamel MS, Wong AKC, Wang Y (2007) Cost-sensitive boosting for classification of imbalanced data. Pattern Recognit 40(12):3358–3378
27. Tan PN, Steinbach M, Kumar V (2005) Introduction to data mining. Pearson, Upper Saddle River

28. Tokunaga T, Iwayama M (1994) Text categorization based on weighted inverse document frequency. In: Special Interest Groups and Information Process Society of Japan, pp 33–39
29. Wang D, Zhang H, Wu W, Lin M (2010) Inverse category frequency based supervised term weighting scheme for text categorization. CoRR arXiv:1012.2609
30. Zheng Z, Wu X, Srihari R (2004) Feature selection for text categorization on imbalanced data. SIGKDD Explor 6(1):80–89

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Murat Okkalioglu received his Ph.D. degree from Anadolu University, Eskisehir, Turkey in 2017. Currently, he is an Assistant Professor in Computer Engineering Department at Yalova University, Turkey. His research interest includes privacy in data mining methods and text classification in general.



Burcu Demirelli Okkalioglu received her B.Sc. degree from Computer Engineering Department at Pamukkale University, Denizli, Turkey, M.Sc. degree from Computer Science at the University of Texas at San Antonio and Ph.D. degree from Anadolu University, Eskisehir, Turkey. Her research interests include data mining, privacy-preserving data mining, and deriving private numeric data from masked data.