Automatic detection of abusive South African tweets using a semi-supervised learning approach

Oluwafemi Oriola[1][0000−0003−0255−6160] and Eduan Kotzé[1][0000−0002−5572−4319]

Department of Computer Science and Informatics, University of the Free State, Bloemfontein, South Africa {OriolaO, kotzeJE}@ufs.ac.za

Abstract. Major setbacks for detection of abusive South African tweets are inadequacy of annotated corpus and high cost of annotation, which semi-supervised learning solves. Semi-supervised learning techniques enhance training data by combining labelled and unlabelled data. However, existing approaches have skewed classification of unlabelled data towards labelled data despite class imbalance of labelled data and unmatched feature distribution between labelled and testing data, that is common in abusive texts. This paper presents a reliable semi-supervised learning approach that reduces the noise in training data by combining features of unlabelled data with varying sizes of important features of labelled data. Chi-square statistics is used for the feature selection, while k-means algorithm is used for clustering of data points. By majority voting rule, reliable labels are assigned to the data points. Classifications with Support Vector Machine and Logistic Regression classifiers show that the proposed approach improves prediction performance.

Keywords: South African Tweets · Abusive Language · Semi-Supervised Machine Learning · Clustering · Classification

1 Introduction

The rise in the act of racial and social media conflicts and their negative consequences means that improved detection of abusive languages on social networks cannot be over-emphasised. One of the major setbacks to improvement of detection of abusive languages in social networks is inadequacy of lexical resources for many languages [1]. Abusive language is referred to as an oral or textual expression that contains dirty words or phrases [2]. This expression can be derogatory, profane, cyber-bullying or hate speech.

Over the last decades, South Africa has experienced upsurge in various degrees of violence such as violent protests and xenophobic attacks, which have led to loss of human and material resources. Many of these violent incidents could be attributed to fast spread of inciteful and abusive comments, perpetrated through social networks. However, there has not been any infrastructural measure to check the soaring volumes of such communications. Recently, South African government promulgated laws to address incidents of hate speech [3], but it is important for such legal tool to be supported by an active preventive measure, which abusive language detection will provide.

As at January 2019, there was twenty-three million active social media users in South Africa, out which twenty percent, about five million have subscribed to Twitter [4]. A recent study of hate speech in multi-domain perspectives has shown that Twitter has been highly used as a medium to propagate racial communications in South Africa [5]. In fact, most of the recent studies on abusive communication online have focused on Twitter. Therefore, this research explores Twitter contents called tweets for detection of abusive language in social media.

Machine Learning is a reliable technique that has been used for abusive tweet detection [6]. Supervised machine learning techniques have been the most widely used but they perform poorly when there are few labelled data. Unsupervised machine learning techniques are used to observe the relationship between features by relying on the similarities among the data and probabilistic approach. They have been mostly applied to zero resourced problems [7]. Some works have combined the two methods as semi-supervised learning, when there are few labelled data and large unlabelled data [8], [9] but they have all relied on deep learning techniques, which are computationally expensive and require huge training data.

South Africa is a multilingual society, but English is mostly used to communicate on the social media. The existing English corpora for abusive language detection have been labelled using crowd-sourcing tools or by annotators that are familiar with the contexts of the abusive discourses. However, studies have shown that language use varies across societies, contexts and individual [10], which is very prominent in abusive South African tweets. Tweets written in English might be code-mixed with profane and non-profane words, in indigenous language or slangs that are peculiar to South Africa. To our best knowledge, the only lexical resources for abusive language that is specific to South Africa is available in Hatebase [11]; however, the resources are very few.

This paper therefore focuses on development of inexpensive and reliable Semi-supervised learning approach, which combines large unlabelled tweets and few labelled tweets to automatically detect abusive tweets.

2 Semi-supervised learning techniques

Semi-supervised learning (SSL) techniques are machine learning techniques that rely on both labelled and unlabelled data for classification tasks. Here, machines learn from fewer labelled data points with the help of large number of unlabelled data points.

Several research works have used unlabelled data to enhance the performance of classification models. These can be categorised into Self-Learning Approach and Active Learning Approach. In Self-learning approach, unlabelled data is automatically annotated and the instances with high confidence are added to the training datasets iteratively. It can be categorised into Self-training methods [12], [13] and Generative learning methods [14], [15]. Active Learning was developed to improve the selection process of unlabelled samples and solve class

imbalance problem, but they often relied on manual method [16] or co-training approach [17], [18], [19], which are costly.

In this work, we are interested in Cluster-then-Label Generative method because they were developed to address missing data problem. Kumar et al. [14] applied Cluster-then-Label to cross-domain adaptation problem, in which unlabelled data in a source domain was merged with unlabelled data in the target domain and clustered using Fuzzy K-Means algorithm. Labels were assigned to the clusters using common knowledge from experts, while classification to predict target dataset was carried out using Dual Margin Binary Hypersphere-based Support Vector Machine. Albalate et al. [15] applied the labelled samples' labels to the clusters of unlabelled data using optimum cluster labelling approach of Hungarian algorithm and removed uncertainty using Silhoutte Cluster Pruning.

Leng et al. [20] proposed Adaptive Semi-supervised clustering algorithm with label propagation to label unlabelled dataset. The available labels of the labelled samples were used to assign labels to the unlabelled data based on K-Nearest Neighbour to core objects defined by adaptive threshold. The adaptive threshold was estimated by the density of each cluster, which the label data point belonged to. Also, new cluster was detected by the distance from the clusters core objects. Peikari et al. [21] clustered labelled and unlabelled datasets and mapped out the high-density regions in the data space. Fuzzy C-Means was used to assign labels to the identified clusters, while Support Vector Machine was used to label the data on the low-density region. Forestier and Wemmert [22] focused on how multiple clustering algorithms can be combined with a supervised learning algorithm to achieve better results than classical semi-supervised and supervised algorithms. They proposed Supervised Learning Ensembles with multiple clustering. The clustering combined labelled and unlabelled objects and maximized intra-cluster similarity using multiple observations.

The above semi-supervised learning approaches have relied on the labels of the labelled data to assign labels to the unlabelled data despite the class-imbalance nature of the labelled data and partially matched features of the labelled and testing data, which is often the case in real-life.

3 Proposed method

In this section, we formalise the approach used to detect South African abusive tweets.

3.1 Semi-supervised learning approach

The Semi-supervised learning method proposed in this work is motivated by the following assumptions which indicate that labelling decision cannot be skewed towards labelled data, in real-life.

- Features of the testing data might not match exactly the features of either labelled or unlabelled data.
- Datasets of similar contexts share asymmetrical features.

3.2 Classification problem

Let X be set of n tweet samples $xi \ni X$. Given a binary-class classification problem with l very low labelled instances and u large unlabelled data such that $U > L$; the set of labelled instances $L = \{(x_1, y_1), \ldots, (x_l, y_l)\}$ and the set of unlabelled instances $U = \{x_{l+1}, \ldots, x_{l+u}\}$, where $y = (0, 1)$ are the class values of the data.

Since the objective of semi-supervised learning is to build a classification model based on the training dataset, then we define our approach as presented in equation (1).

$$y = CX(x) : y\epsilon\{0, 1\} \tag{1}$$

The schematic diagram in Figure 1 depicts the semi-supervised learning process. This involves three procedures: labelling of unlabelled data as described in section 3.3 and 3.4, training of merged unlabelled and labelled data, and testing with test data as described in section 4.3.3.

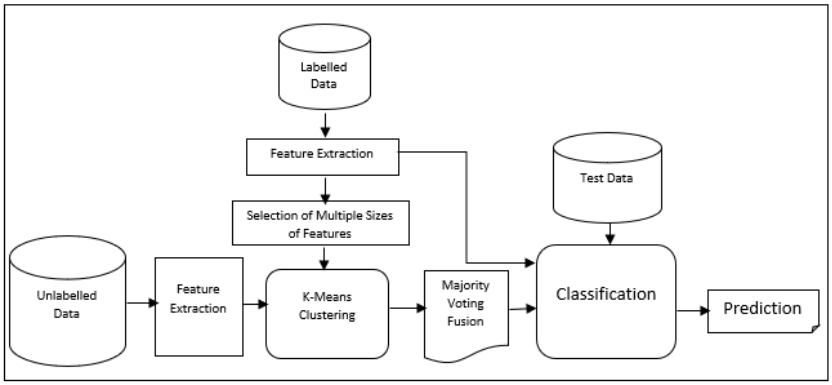

Fig. 1. Proposed Semi-supervised Learning Process

3.3 Clustering

In order to apply both unlabelled and labelled data samples as training data without skewness towards labelled data, the features of the unlabelled data are fused with different sizes of features from labelled data. Given that R is the set of features of unlabelled data U and S is the set of features of labelled data, that is $R = \{r_l + 1, \ldots, r_l + u\}$ and $S = \{s_1, \ldots, s_l\}$ Then, $L\epsilon\{a_1, a_2, \ldots, a_q\}$ and $a_i < a_{i+1}$ where a ϵA is the linearly selected size of features and q is the size of features.

By Matrix Multiplication,

$$R^0 = R * S \tag{2}$$

Applying K-Means algorithm to cluster partition of R^o into k disjoint clusters $C = \{0, 1\}$ given that $k - y_i = 0$, we get J sets of cluster partitions $(j_1, j_2, \ldots, j_i)$

$$j_i = argmin_t \sum_{i-1}^{k} \sum_{r \epsilon ti} \left|\left|R^0 - \mu i\right|\right| \tag{3}$$

The pseudocode is presented below:

Data: Labelled data features (R); unlabelled data instances (U)
Result: The labels VT for unlabelled data instances y for qi
for $R = L + 1, L + 2, \ldots, L + u$ **do**
  for $qi = 3, 5, 10, 15, 20, 25, 30$ **do**
    for $S = 1, 2, \ldots, L$ **do**
      $R^0 = R * S$;
      $j = argmin_t \sum_{i-1}^{k} \sum_{r \epsilon ti} \left|\left|R^0 - \mu i\right|\right|$;
      $V_T = j$;
    end
  end
end

3.4 Fusion and Labelling

n order to assign labels to the unlabelled data, majority vote rule is applied to every instance of R0 in A such that $V_T > 0$.

The selected label

$$y_i = C_u(max(V_t)) \tag{4}$$

Where $R0 = \{r_{l+1}, \ldots, r_{l+u}\}$, V_T = labels for each instance for all tested sizes of features.

The pseudocode is presented below:

Data: Labels for unlabelled data instances, V_T
Result: Majority Voting labels, V_m
for $T = 1, \ldots, t$ **do**
  max count = Max (AllCounters)
  if *max count* $> t/2$ **then**
    V_m =class label corresponding to max count
  end
end

4 Experimental steps

4.1 Data collection and annotation

Total of 21,350 tweets of South African discourses on Twitter between the period of May 5, 2019 and May 13, 2019 were collected using Twitter Archival tool, a Google Sheets plugin, that works based on Twitter Search API. The collection

targeted tweets related to 2019 South African national elections, popular South Africa individuals and trending issues such as land reclamation, Orania and white communities. Tweets that contained non-English words were removed, except names of individuals, towns, people, and organizations. Retweets and repeated tweets as well as tweets with empty word characters were also removed. Following these steps, the number of tweets remaining was 10,245. The tweets were divided into three data samples of labelled, unlabelled and testing data. Total of 1,737 tweets were randomly selected for annotation while the remaining 8,548 tweets were not annotated. The selected samples were annotated by two annotators as either 'abusive' (A) or 'non-abusive' (NA), from which 1,730 tweets were selected because of agreement on their labels. The Cohen's Kappa agreement score [23] was 0.8490, which indicated almost excellent agreement. The 7 tweets, which were disagreed upon were added to the unlabelled dataset thus, making it 8,555. The 1,730 tweets were divided into 338 labelled tweets and 1,352 testing data. The resulting distribution of the dataset is presented in Table 1.

Table 1. Distribution of Dataset

Dataset	Samples	Number of Instances	Non-abusive (NA)	Abusive (A)
Training	Unlabelled data	8555	-	-
	Labelled data	338	286	52
Testing	Testing data	1352	1118	234

4.2 Data pre-processing

The samples of the dataset went through various stages of pre-processing so as to be suitable for text processing. These stages include removal of username, punctuations, special characters and symbols including emoticons and emojis, removal of hash symbols in hashtags, removal of English stopwords, stemming and change of all texts to lower case.

4.3 Data processing

Three major stages of process were involved, which include feature engineering, clustering and classification.

Feature Engineering The texts in the tweets were transformed into Term Frequency and Inverse Document Frequency (TF-IDF) feature space, where weights were created as indicated in equation (5) [24]. TF-IDF was chosen over Bag of

Words (BoW) because TF-IDF considers the IDF of each term unlike BoW and performed better than most surface-level feature representations [25]. The TF-IDF weights for a given term t in a document d is given as:

$$TF - IDF(t, d) = TF(t, d) * IDF(t) \tag{5}$$

when $IDF(t) = log[n/(DF(t) + 1]$, n = total number of documents in the document set; $DF(t)$ = document frequency of t We also made use of word and character n-gram models. The word n-gram include as Unigram (1), combination of Unigram and Bigram (1, 2) and combination of Unigram, Bigram and Trigram (1, 2, 3), while character n-gram include character n-gram with length sizes from 2 to 6 (2-to-6), 3 to 7 (3-to-7) and 4 to 8 (4-to-8)

Clustering We employed TF-IDF vectorization on the features of the labelled and unlabelled data, without over or under-sampling. The important features of the labelled samples were extracted using the Chi-Square statistics [26] with K values of 3, 5, 10, 15, 20, 25 and 30. This was followed by fusion of the features of labelled and unlabelled samples as described in equation (4). The K-Means unsupervised learning algorithm (number of clusters = 2) was used to cluster the fused samples, resulting in seven different cluster samples of two cluster partitions each. By majority voting rule presented in equation (4), the most reliable cluster partition was obtained. Abusive label was assigned to cluster partition with more abusive words, while non-abusive label was assigned to cluster partition with lesser abusive words.

Classification We applied n-gram features weighted by TF-IDF vectorization on the combination of semi-supervised labelled data and the originally labelled data samples. They were used because of their effective performance in previous text classification problems [5], [25]. SMOTE oversampling technique [24] was applied to reduce class imbalance. The testing data sample was also transformed in the same manner. Support Vector Machine (kernel=linear kernel) and Logistic Regression (kernel=liblinear) classifiers were used to train the merged data samples and detect abusive tweets from the testing data sample using the n-gram features.

In order to select the best training model, Grid-search Hyperparameter Tuning approach was implemented over 10-fold Cross Validation. For the Support Vector Machine classifier (SVM), different C-regularization values ranging from 0.001 to 1000 were tested. For the Logistic Regression (LogReg), both L1 and L2 penalty functions with np.logspace values over -4, 4 and 20 were tested.

4.4 Performance metrics

Precision, Recall, F-Measure, Accuracy and Mean Accuracy are metrics used to evaluate the performance of the proposed Semi-supervised Learning Method. The performance metrics are defined as presented in equations (6) to (10). The

equations rely on the true positive (TP), which is the number of correctly predicted abusive tweets; true negative (TN), which is the number of correctly predicted non-abusive tweets; false positive (FP), which is the number of incorrectly predicted abusive tweets; false negative (FN), which is the number of incorrectly predicted non-abusive tweets.

$$\text{Precision P} = \frac{TP}{TP + FP} \tag{6}$$

$$\text{Recall R} = \frac{TP}{TP + FN} \tag{7}$$

$$\text{F-Measure F1} = \frac{2X(Recall\ X\ Precison)}{(Recall + Precision)} \tag{8}$$

$$\text{Accuracy A} = \frac{TP + TN}{TP + FP + TN + FN} \tag{9}$$

$$\text{Mean Accuracy} = \frac{\sum_{i=1}^{10} Ai}{10} \tag{10}$$

4.5 Performance evaluation

The proposed semi-supervised learning approach (SSL) was compared with two supervised learning approaches.

- **Method A**: This is a supervised learning method, in which only labelled data was used for training data.
- **Method B**: This is a supervised learning method with unmatched training data distribution consisting of labelled data and unlabelled data, which pseudo-labels was obtained by K-means clustering of unlabelled data without the features of the labelled data.

5 Results

We evaluated the SSL for word n-gram and character n-gram features with Support Vector Machine and Logistic Regression classifiers and compared the performances with method A and method B. The testing results of accuracy and mean accuracy for SVM and LogReg over 10-fold cross validation are presented in Table 2 and Table 3, respectively. Also, the results of Precision, Recall and F-Measure for SVM for word and character n-gram features are presented in Table 4, while the results of LogReg are presented in Table 5. The comparisons of accuracy and F-measure of the SSL, method A and method B are presented in Figure 2.

The results of the performance of the SSL, method B and method A in Table 2 showed that the SSL recorded the highest accuracy of 0.9585 and 0.9667 for word n-gram and character n-gram, respectively followed by method A. The difference

Table 2. Accuracy and Mean Accuracy for Support Vector Machine testing

Model	Feature Type	Best Features (n-gram)	Best Parameter	Accuracy	Mean Accuracy
Method A	Word	1	C = 1	0.8335	0.8727
	Char	3-to-7	C=1	0.8713	0.8816
Method B	Word	1,2,3	C=0.01	0.7988	0.8639
	Char	4-to-8	C=0.001	0.7980	0.8639
SSL	Word	1	C=0.01	0.9585	0.9581
	Char	2-to-6	C=1000	0.9667	0.9597

in the accuracy of the SSL and method A were 0.0942 (≈0.1) and 0.0968 (≈0.1) for word and character n-gram, respectively. In Table 3, the SSL recorded the highest accuracy of 0.9578 and 0.9696 for word n-gram and character n-gram, respectively followed by method A. The difference in the accuracy of the SSL and method A were 0.1228 and 0.0924 (≈0.1) for word and character n-gram, respectively. The same ranges of differences were recorded in mean accuracy, which showed clearly that the SSL convincingly outperformed method A and method B, in terms of accuracy. Method B recorded the lowest accurcy in both Table 2 and Table 3.

Table 3. Accuracy and Mean Accuracy for Logistic Regression Testing

Model	Feature Type	Best Features (n-gram)	Best Parameter	Accuracy	Mean Accuracy
Method A	Word	1,2	L1, C = 3792.69	0.8350	0.8786
	Char	3-to-7	L2, C= 78.47	0.8772	0.8816
Method B	Word	1	L1, C = 0.0001	0.7980	0.8639
	Char	2-to-6	L1, C = 0.0001	0.7980	0.8639
SSL	Word	1	L1, C = 29.76	0.9578	0.9580
	Char	2-to-6	L2, C = 78.47	0.9696	0.9613

In Table 4, the value of precision, recall and F-Measure for SSL were higher than both method A and method B. While 0.96 and 0.95 precisions were recorded

Table 4. Precision, Recall and F-Measure for Support Vector Machine testing

Model	Feature Type	Precision		Recall		F-Measure	
		NA	A	NA	A	NA	A
Method A	Word	0.89	0.59	0.91	0.53	0.9	0.56
	Char	0.89	0.76	0.96	0.52	0.96	0.62
Method B	Word	0.8	0.8	1	0	0.89	0.01
	Char	0.8	0	1	0	0.89	0
SSL	Word	0.96	0.95	0.99	0.8	0.98	0.87
	Char	0.97	0.94	0.99	0.87	0.98	0.9

Table 5. Precision, Recall and F-Measure for Logistic Regression Testing

Model	Feature Type	Precision		Recall		F-Measure	
		NA	A	NA	A	NA	A
Method A	Word	0.88	0.6	0.92	0.49	0.9	0.54
	Char	0.89	0.8	0.97	0.51	0.93	0.62
Method B	Word	0.8	0	1	0	0.89	0
	Char	0.8	0	1	0	0.89	0
SSL	Word	0.98	0.87	0.97	0.89	0.97	0.88
	Char	0.98	0.93	0.99	0.89	0.98	0.91

by the SSL in word n-gram evaluation for non-abusive and abusive, respectively, method A recorded 0.80 and 0.80, and method B recorded 0.89 and 0.90. In character n-gram evaluation, the SSL recorded 0.97 and 0.94, respectively, method A recorded 0.89 and 0.76, while method B recorded 0.80 and 0.00. In the case of recall, the highest values of 0.99 and 0.80 were recorded by the SSL for non-abusive and abusive, respectively, followed by 0.91 and 0.53 for method A. In character n-gram, the SSL recorded the highest recall of 0.99 and 0.87, respectively, followed by method A. The 1.00 recall against 0.00 for method B showed bias against abusive class during training. In the case of F-Measure, the SSL recorded the higest performance of 0.87 and 0.90 for abusive tweet detection for word n-gram and character n-gram, respecively followed by method A. The value of precision, recall and F-Measure for the SSL in Table 5 were also higher than both method A and method B.

In character n-gram evaluation, the SSL recorded 0.98 and 0.93, respectively, method A recorded 0.89 and 0.80, while method B recorded 0.80 and 0.00. For recall, while 0.97 and 0.89 precision was recorded by proposed SSL for non-abusive and abusive, respectively, method A recorded 0.92 and 0.49 and method B recorded 1.00 and 0.00. For character n-gram, the SSL recorded 0.99 and 0.89, respectively, method A recorded 0.96 and 0.52, while method B recorded 1.00 and 0.00. The 1.00 recall against 0.00 for method B, showed bias against abusive class during training. In terms of F-Measure, the SSL recorded the higest performance of 0.90 and 0.91 for abusive tweet detection for word n-gram and character n-gram respecively followed by method A. The lowest F-Measure was recorded by method B.

The bar chart in Figure 2(a) showed that there was consistent increase in the width of the charts from method B through method A to the SSL. There was also slight increase from word n-gram to character n-gram. In Figure 2(b), drastic rise was observed from the bar of method B to method A. So also was drastic rise from method A to proposed SSL. These outcomes indicated that the proposed SSL addressed the problems of class imbalance and unmatched distribution.

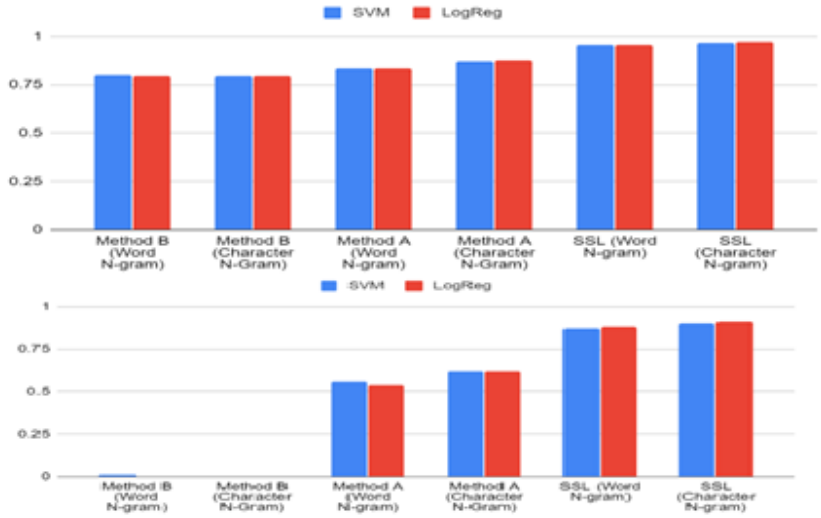

Fig. 2. Performance for the Different Methods : (a) Accuracy (b) F-measure

6 Conclusion

We have developed a semi-supervised learning approach that combined both labelled and unlabelled data, without skewness towards labelled data for improved detection of abusive tweets in binary classification model. The approach reduced the impact of class imbalance and unmatched distribution among labelled, unlabelled and testing data features.

Matrix multiplication was used to fuse the labelled and unlabelled features; K-Means algorithm was used to cluster the fused features; majority voting rule was applied to select reliable labels for the unlabelled samples. The labelled and the previous unlabelled samples were used as training data. The performance of the approach was evaluated using word n-gram and character n-gram features as well as support vector machine and logistic regression classifiers. The results showed that our semi-supervised learning approach performed better than supervised learning approaches, with few training data or noisy training data. In future, classification of abusive language will be considered.

References

1. A. Søgaard, I. Vulic, S. Ruder, and M. Faruqui. Cross-lingual word embeddings. *Synth. Lect. Hum. Lang. Technol*, page 132 1–132.
2. M.O. Ibrohim and I. Budi. Sciencedirect a dataset dataset and and preliminaries preliminaries study study for for abusive abusive language language detection detection in indonesian social media in indonesian social media. *Procedia Comput. Sci*, 135:222–229.
3. R.S.A. Republıc of south afrıca preventıon and combatıng of hate crımes and hate.
4. J. Clement. South africa: digital population as of january 2019. *Statista*. Available.
5. Multilingual cross-domain perspectives on online hate speech. Clips tech. rep. series 8,.
6. A. Schmidt and M. Wiegand. A survey on hate speech detection using natural language processing.
7. H. Kamper, L. Karen, and G. Sharon. An embedded segmental k-means model for unsupervised segmentation and clustering of speech. *Comput.* Lang. arXiv:1703,.
8. C. Khatri, B. Hedayatnia, R. Goel, A. Venkatesh, R. Gabriel, and A. Mandal. Detecting offensive content in open-domain conversations using two stage semi-supervision arxiv : 1811. *12900v1 [cs*, 30(v 2018).
9. I. Gunasekara. A review of standard text classification practices for multi-label toxicity identification of online content.
10. B.language Norton. Identity and the ownership of english. *Language and Identity*, 31(3):409–429.
11. HateBase. Hatebase: The world's largest structured repository of regionalized, multilingual hate speech. *Hatebase*. Available at:.
12. I.E. Livieris, A. Kanavos, V. Tampakas, and P. Pintelas. An auto-adjustable semi-supervised.
13. B.A. Hamilton. Semi-supervised learning with self-supervised networks. arXiv:1906.10343v1.
14. S. Kumar, X. Gao, and I. Welch. Cluster-than-label : Semi-supervised approach for domain adaptation. In *2017 IEEE 31st International Conference on Advanced Information Networking and Applications*, page 704–711.
15. A. Albalate, A. Suchindranath, D. Suendermann, and W. Minker. A semi-supervised cluster-and-label approach for utterance classification. In *Interspeech2010*, page 1–4.
16. M. Chegini. *clustering , and active learning*, 3:9–17.
17. D.-H. Lee. Pseudo-label : The simple and ecient semi-supervised learning method for deep neural networks. In *ICML 2013 Workshop : Challenges in Representation Learning (WREPL*, page 2–7. Atlanta, Georgia, USA.
18. B. Miller. Active learning approaches for labeling text.
19. Y. Li, Y. Lv, S. Wang, J. Liang, J. Li, and X. Li. Cooperative hybrid semi-supervised learning for text classification. *Symmetry*, 11(2).
20. Mingwei Leng, Jinjin Wang, Jianjun Cheng, and X. Hanhai Zhou. C. *Journal of Software. J. Softw. Eng*, 22:1–7.
21. M. Peikari, S. Salama, S. Nofech-mozes, and A.L. Martel. Open a cluster-then-label semi- supervised learning approach for pathology image classification. *Sci. Rep*, 1–13.
22. G. Forestier, C. Wemmert, G. Forestier, and C. Wemmert. Semi-supervised learning using multiple clusterings with limited labeled data with limited labeled data. *Inf. Sci. Elsevier*, 361–362:48–65.

23. J. Cohen. A coefficient of agreement for nominal scales. *Educ. Psychol. Meas*, 20:37–46.
24. Pedregosa F., Varoquaux G., Gramfort A., Michel V., Thirion B., G. O., and D.E. Scikit-learn. Machine learning in python. *J. Mach. Learn. Res*, 12:2825–2830.
25. T. Davidson, D. Warmsley, M. Macy, and I. Weber. Automated hate speech detection and the problem of offensive language. In *Proceedings of the Eleventh International AAAI Conference on Web and Social Media(ICWSM*, page 512–515.
26. J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: A comprehensive study. *Int. J. Comput. Vis.*