Lifelong Learning of Hate Speech Classification on Social Media

Jing Qian[†], **Hong Wang**[†], **Mai ElSherief**[*], **Xifeng Yan**[†]
[†] University of California, Santa Barbara
[*] Georgia Institute of Technology
{jing_qian,hongwang600,xyan}@cs.ucsb.edu
mai.h.sherief@gmail.com

Abstract

Existing work on automated hate speech classification assumes that the dataset is fixed and the classes are pre-defined. However, the amount of data in social media increases every day, and the hot topics changes rapidly, requiring the classifiers to be able to continuously adapt to new data without forgetting the previously learned knowledge. This ability, referred to as lifelong learning, is crucial for the real-word application of hate speech classifiers in social media. In this work, we propose lifelong learning of hate speech classification on social media. To alleviate catastrophic forgetting, we propose to use Variational Representation Learning (VRL) along with a memory module based on LB-SOINN (Load-Balancing Self-Organizing Incremental Neural Network). Experimentally, we show that combining variational representation learning and the LB-SOINN memory module achieves better performance than the commonly-used lifelong learning techniques.

1 Introduction

With the rapid rise in user-generated web content, the scale and complexity of online hate have reached unprecedented levels in recent years. ADL (Anti-Defamation League) conducted a nationally representative survey of Americans in December 2018 and the report shows that over half (53%) of Americans experienced some type of online harassment.[1] This number is higher than the 41% reported to a comparable question asked in 2017 by the Pew Research Center (Center, 2017). To address the growing online hate, a great deal of research has focused on automatic hate speech classification. Most of the previous work focuses on binary classification (Warner and Hirschberg, 2012; Zhong et al., 2016; Nobata et al., 2016; Gao et al., 2017; Qian et al., 2018b) or coarse-grained multi-class classification (Waseem and Hovy, 2016; Badjatiya et al., 2017; Davidson et al., 2017). Qian et al. (2018a) argue that fine-grained classification is necessary for fine-grained hate speech analysis. The Southern Poverty Law Center (SPLC) monitors hate groups throughout the United States by a variety of methodologies to determine the activities of groups and individuals, including reviewing hate group publications.[2] Therefore, instead of differentiating normal posts from the other offensive ones, Qian et al. (2018a) propose a more fine-grained hate speech classification task that attributes hate groups to individual tweets. However, a common limitation of all the research mentioned above is that they assume the dataset to be static and train the classifiers on each isolated dataset, i.e., isolate learning, ignoring the rapid increase of the amount of data in social media and the rapid change of the hot topic.

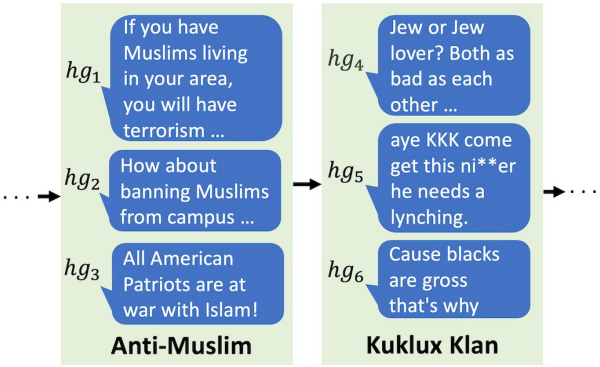

Figure 1: An illustration of our proposed task. hg_i: the ith hate group. The model is trained on a sequence of sub-datasets, split by their hate ideologies, e.g., anti-Muslim and Kuklux Klan. The task on each sub-dataset is to identify the hate group given the tweet.

A report from L1ght[3], a company that specializes in measuring online toxicity, suggests that

[1] https://www.adl.org/onlineharassment

[2] https://www.splcenter.org/fighting-hate/extremist-files/ideology

[3] https://l1ght.com/Toxicity_during_coronavirus_Report-L1ght.pdf

Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 2304–2314
June 6–11, 2021.

amid the growing threat of the coronavirus, there has been a 900% growth in hate speech towards China and Chinese people on Twitter since February 2020. As a result of the rapid change of social media content, the hate speech classifiers are required to be able to continuously learn and accumulate knowledge from a stream of data, i.e., lifelong learning. Learning on each portion of the data is considered as a task, so a stream of tasks are joined to be trained sequentially. In this work, we propose a novel lifelong fine-grained hate speech classification task, as illustrated in Figure 1. The models trained by isolate learning tend to face catastrophic forgetting (McCloskey and Cohen, 1989; Ratcliff, 1990; McClelland et al., 1995; French, 1999) due to a non-stationary data distribution in lifelong learning. To address this problem, an extensive body of work has been proposed for various lifelong learning tasks. However, our experiments show that the commonly-used lifelong learning methods still exhibit catastrophic forgetting in our proposed tasks. One important difference between the Twitter hate group dataset and the other image datasets commonly used in lifelong learning study is that the similarity among the different tasks is unstable and relatively low, as indicated by the low average Jaccard Indexes of the topic words in Table 1. To alleviate this problem, we introduce VRL to distill the knowledge from each task into a latent variable distribution. We also augment the model with a memory module and adapt the clustering algorithm, LB-SOINN, to select the most important samples from the training dataset of each task.

Our contributions are three-fold:

- This is the first paper on lifelong learning of fine-grained hate speech classification.
- We propose a novel method that utilizes VRL along with an LB-SOINN memory module to alleviate catastrophic forgetting resulted from a severe change of data distribution.
- Experimental results show that our proposed method outperforms the state-of-the-art significantly on the average F1 scores.

2 Related Work

Most research on lifelong learning alleviates catastrophic forgetting in the following three directions. **Regularization-based Methods:** These methods impose constraints on the weight update. The goal of the constraints is to minimize deviation from trained weights when training on a new task. The constraints are generally modeled by additional regularization terms (Kirkpatrick et al., 2017; Zenke et al., 2017; Fernando et al., 2017; Liu et al., 2018; Ritter et al., 2018). Elastic Weight Consolidation (EWC) (Kirkpatrick et al., 2017) alleviates catastrophic forgetting by slowing down learning on the model parameters which are important to the previous task. The importance of the parameters is estimated by the Fisher information matrix. Instead of the Fisher information matrix, PathNet (Fernando et al., 2017) uses agents embedded in the neural network to determine which parameters of the neural network can be reused for new tasks and the task-relevant pathways are frozen during training on new tasks.

Architecture-based Methods: The main idea of this approach is to change architectural properties to dynamically accommodating new tasks, such as assigning a dedicated capacity inside a model for each task. Rusu et al. (2016) propose Progressive Neural Networks, where the model architecture is expanded by allocating a new column of neural network for each new task. Part and Lemon (2016, 2017) combine Convolutional Neural Network with LB-SOINN for incremental online learning of object classes. Although they also use LB-SOINN in their work, the usage of LB-SOINN in this work is completely different. They use LB-SOINN to predict object class while our proposed method adapts the original LB-SOINN to calculate the importance

Ideology	Avg. JI	Keywords
Christian Identity	0.019	Jesus, Yahuwshua
Radical Tr. Catholic	0.031	catholic, remnant
Neo Confederate	0.039	southern, Free Dixie
Anti Semitism	0.047	Israel, Trump
Anti Catholic	0.049	Texe Marrs, truth
Hate Music	0.049	death, radio
Anti Muslim	0.064	Muslim, Islam
Black Separatist	0.071	black, panther
Racist Skinhead	0.074	shirt, white
Anti Immigration	0.075	immigration, border
Holocaust Identity	0.078	Jewish, Trump
Neo Nazi	0.091	Hitler, white
Kuklux Klan	0.100	ni**a, f**king
Anti LGBTQ	0.100	family, marriage
White Nationalist	0.105	white, America

Table 1: Information about the 15 hate ideologies. Tr.: Traditional. Avg JI: the average of the Jaccard Index between the topic words of one ideology and those of another ideology. The topic words are extracted by Latent Dirichlet Allocation (LDA) (Blei et al., 2003). The top 2 most frequent topic words are selected as keywords.

of the training samples without making any prediction on the class. A problem with the methods in this category is that the available computational resources are limited in practice. As a result, the model expansion will be prohibited when the number of tasks increases to a certain degree.

Data-based Methods: These methods alleviate catastrophic forgetting by utilizing a memory module, which either stores a small number of real samples from previous tasks or distills knowledge from previous tasks. The main feature of Gradient Episodic Memory (GEM) (Lopez-Paz and Ranzato, 2017) is the episodic memory, storing a subset of the samples from the observed tasks. GEM computes the losses on the episodic memories and treats them as inequality constraints, avoiding them to increase. Averaged GEM (Chaudhry et al., 2019) is a more efficient version of GEM. de Masson d'Autume et al. (2019) propose a lifelong language learning model using a key-value memory module for sparse experience replay and local adaptation. Sun et al. (2020) formulate lifelong language learning as a language modeling task and replay the generated pseudo-samples of previous tasks during training.

There are also studies combining multiples methods above. Xia et al. (2017) combine the architecture-based method and the data-based method. Wang et al. (2019) combine the regularization method and the data-based method for lifelong learning on relation extraction. Our proposed method is also a combination of the regularization method and the data-based method but in a different way.

3 Task Description

We use the dataset as in Qian et al. (2018a), where the tweet handles are collected based on the hate groups identified by SPLC. SPLC categorizes these hate groups according to their hate ideologies. For each hate ideology, the top three Twitter handles are selected in terms of the number of followers. The dataset includes all the content (tweets, retweets, and replies) posted with each Twitter account from the group's inception date, as early as 2009, until 2017. Altogether, the dataset consists of 42 hate groups from 15 different ideologies. Table 1 shows the 15 ideologies. Each instance in the dataset is a text tuple of (tweet, hate group name, hate ideology).

We separate the dataset by ideology. The reason is that various existing hate speech datasets collect data using keywords or hashtags (Waseem and Hovy, 2016; Davidson et al., 2017; Golbeck et al., 2017), which have a strong relationship with hate ideologies or topics. We also observe that the hot spots of society can lead to a significant shift of major hate speech topics or the emergence of new hate ideologies on social media as mentioned in section 1, indicating that the expansion of the hate speech dataset may be accompanied by the emergence of new hate ideologies.

Therefore, we separate the collected data into a sequence of 15 subsets according to their ideologies and sort them by the date of the first tweet post in each subset, from the earliest to the latest. The task on each subset is to identify the hate group given the tweet text. Qian et al. (2018a) propose a hierarchical Conditional Variational Autoencoder model for the fine-grained hate speech classification task. The architecture and the training process of their model require the number of classes to be pre-defined. However, we do not pre-define the number of classes in our task since such kind of information is not available in the real-world application of lifelong learning. The model should be able to incorporate emerging hate groups at any time of training. In order to satisfy this condition, we formulate the task of identifying the group as a ranking task, instead of a classification task. For each tweet, we provide the model with a set of candidate groups, consisting of all the previously seen hate groups, including the ground truth group. The model takes each combination of the tweet and the candidate group as input and outputs a score. The corresponding loss function is:

$$\mathcal{L}_r = \sum_{(x,y_s)\in D} \sum_{y_i \in Y\setminus\{y_s\}} h(f_\theta(x,y_s) - f_\theta(x,y_i)) \quad (1)$$

where x is the tweet text, y_s is the ground truth group of x. Y is candidate group set of x, which consists of all the seen hate groups until x is observed by the model, including the ground truth group y_s of x, so $y_i \in Y\setminus\{y_s\}$ is the negative candidate group of x. f_θ is the scoring model parameterized by θ. $h(a) = \max(0, m-a)$, m is the chosen margin.

Same as in other lifelong learning studies, we consider learning on each of the hate ideologies in the sequence as a task, so we have a sequence of 15 tasks. As mentioned in section 1, the similarity among our tasks is unstable and relatively low.

Therefore, when the model is continuously trained on the tasks, it may encounter a sudden change of vocabulary, topic, and input data distribution. This makes our tasks more challenging compared to the other lifelong learning tasks because the abrupt change can make the catastrophic forgetting problem more severe. This is also the reason that some techniques achieving significant improvement in the image classification tasks do not perform well on our task (see section 5).

4 Our Approach

As mentioned in section 2, one way to alleviate catastrophic forgetting is to use a memory module, storing a small number of real samples from previous tasks and a simple way to utilize the memorized samples is to replay the memory when training on a new task, such as mixing them with the training samples from the current task. The idea behind this approach is that the memorized samples should reflect the data distribution so that the replay of the memory can help the model make invariant predictions on the samples of the previous tasks. However, this approach may not work well when the size of the memory is small. The reason is that when there is only a small amount of data memorized, the memory is not able to reflect the data distribution of the previous task and thus the model can easily overfit on the memorized samples instead of generalizing to all the samples in the previous task.

We address this problem from two aspects. First, since the memory size is limited, it is beneficial to select the most representative training samples in the previous tasks to memorize. Second, simply storing the real training samples in the memory may not be sufficient to represent the knowledge of the previous tasks, so we need a better way to distill knowledge from the observed samples along with a method to utilize it when training on a new task. We combine two techniques: Variational Representation Learning (VRL) and Load-Balancing Self-Organizing Incremental Neural Network (LB-SOINN) to achieve these goals. We propose a supervised version of LB-SOINN to select the most important training samples in the current task. VRL not only distills the knowledge from the current training task but also provides an appropriate hidden representation as input for the LB-SOINN, so we introduce VRL first.

4.1 Variational Representation Learning

The distilled knowledge of previous tasks can take various forms, but the key point is that it should be related to the data distribution of the corresponding task so that it can be utilized to alleviate catastrophic forgetting. Inspired by the Variational Autoencoder (VAE) (Kingma and Welling, 2013), we consider the distribution of the hidden representation of the input data as the distilled knowledge.

The original VAE model is proposed for data generation, so the objective of the original VAE is:

$$Obj = \sum_{x \in X} \log p(x) \tag{2}$$

$$p(x) = \int_z p(x|z)p(z)dz \tag{3}$$

z is the latent variable, i.e., the hidden representation of the input. Since the integration over z is intractable, we instead try to maximize the corresponding evidence lower bound (ELBO) and the corresponding loss function is as follows:

$$\mathcal{L}_{vae} = \sum_{x \in X} E_{z \sim p_\alpha(z|x)}[-\log p_\varphi(x|z)] + D_{KL}[q_\alpha(z|x)||p_\beta(z)] \tag{4}$$

$p(x|z)$, $q(z|x)$, and $p(z)$ are the likelihood distribution, posterior distribution, and prior distribution. α,φ, and β indicate parameterization. The loss function can be separated into two parts. The first part $E[-\log p(x|z)]$ is the reconstruction loss, trying to reconstruct the input text from the latent variable. It pushes z to reserve as much information of the input as possible. This is consistent with our goal to learn the knowledge of the data distribution. The second part is $D_{KL}[q(z|x)||p(z)]$, where D_{KL} is the Kullback–Leibler (KL) divergence. Minimizing it pushes the posterior and the prior distributions to be close to each other. By assuming the posterior $p(z|x)$ to be a multivariate Gaussian distribution $\mathcal{N}(\mu_z, \Sigma_z)$, the latent variable z is sampled from $\mathcal{N}(\mu_z, \Sigma_z)$.

In the original VAE, $p(z)$ is chosen to be a simple Gaussian distribution $\mathcal{N}(0, 1)$. However, this is over-simplified in our task because different from the unsupervised generation task of the original VAE, our ranking task is supervised. Our task not only requires z to contain information of the tweet text itself but also requires it to indicate the group information of the tweet. In other words, the distilled distribution should be conditioned on both the

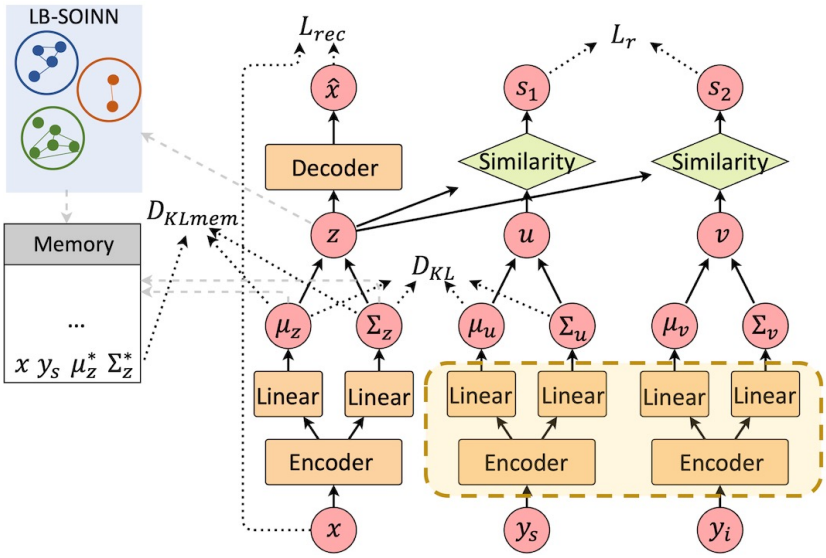

Figure 2: An illustration of our method. The dotted arrows indicate the computation of the loss. The light-colored dashed arrows illustrate the update of the memory module. Note that the layers in the rounded rectangle share parameter weight. There is only one encoder for the group input, followed by two linear layers. We make a copy of it in the figure just for a clear illustration of loss computation. $\hat{x}$: the reconstructed tweet input. s_1, s_2: scores of (x, y_s) and (x, y_i) separately. μ_z^* and Σ_z^* are the previously memorized distribution on the latent variable of x. L_{rec} is the reconstruction loss, which is the first term in equation 4. Please refer to section 4 for the meaning of other variables in the figure.

tweet and its group label to reflect the data distribution in a supervised task. Setting the prior to be the same for all the hate groups pushes z or the distribution of z to ignore the label information. Instead, the prior should be different for each hate group, so we replace $p(z)$ with $p(u|y_s)$, where y_s is the group label of x and u is the latent variable. $p(u|y_s)$ is assumed to be a multivariate Gaussian distribution $\mathcal{N}(\mu_u, \Sigma_u)$. Note that the replacement itself can not guarantee $p(u|y_s)$ to be different for each hate group because the loss function in equation 4 does not push $p(u|y_s)$ to satisfy this condition. However, the ranking loss function 1 fills in the gap. Therefore, our loss function on the current training task is a combination of these two.

$$\begin{aligned}\mathcal{L}_{cur} = &\sum_{(x,y_s)\in D}\sum_{y_i\in Y\setminus\{y_s\}} h(f_\theta(x,y_s) - f_\theta(x,y_i)) \\ &+ E_{z\sim p_\alpha(z|x)}[-\log p_\varphi(x|z)] \\ &+ D_{KL}[q_\alpha(z|x)||p_\beta(u|y_s)]\end{aligned} \tag{5}$$

The right part of Figure 2 illustrates the computation process of VRL.

4.2 LB-SOINN Memory Module

VRL provides a way to summarize knowledge into latent variable distributions. However, we still need a method to utilize the learned distribution to alleviate catastrophic forgetting. We do this by incorporating a memory module D_{mem} to store a small subset of important training samples along with their latent variable distributions, so each sample stored in the memory is a tuple of $(x, y_z, q_{\alpha'}(z|x))$. Here $q_{\alpha'}(z|x)$ is the distribution computed when the model completes training on the task that (x, y_z) belongs to. The memorized samples are taken as anchor points when training on a new task. We introduce a memory KL divergence loss to push $q_\alpha(z|x)$ computed when training on a new task to be close to the memorized distribution $q_{\alpha'}(z|x))$. Therefore, the complete loss function is:

$$\begin{aligned}\mathcal{L} &= \mathcal{L}_{cur} + D_{KLmem} \\ &= \mathcal{L}_{cur} + \sum_{(x,y_s)\in D_{mem}} D_{KL}[q_\alpha(z|x)||q_{\alpha'}(z|x))]\end{aligned} \tag{6}$$

Since the size of the memory is limited, we introduce a supervised version of LB-SOINN to select the most important training samples in the current task. The input for the LB-SOINN is the hidden representation of the tweet text, which is z in the case of Variational Representation Learning (see Figure 2). We refer readers to Zhang et al. (2013) for the detailed explanation of LB-SOINN. The original LB-SOINN is an unsupervised clustering algorithm that clusters unlabeled data by topology

learning. We utilize the topology learning of LB-SOINN instead of clustering since our task is supervised. Therefore, we make the following adjustments to the original LB-SOINN.

1) The criteria to add a new node: Add a new node to the node set if one of the following condition is satisfied: a) The distance between the input and the winner is larger than the winner's threshold. b) The distance between the input and the second winner is larger than the second winner's threshold. c) The label of the input sample is not the same as the label of the winner.

2) Build connections between nodes: Connect the two nodes with an edge only if the winner and the second winner belong to the same class.

3) We disable the removal of edges whose ages are greater than a predefined parameter. We disable the deleting of nodes and the algorithm of updating the subclass labels of every node. The node label is the label of the instances assigned to it. Our adjusted algorithm guarantees that each node will only be assigned the samples from one class.

LB-SOINN keeps track of the density of each node, which is defined as the mean accumulated points of a node. A node gets points when there is an input sample assigned to it. If the mean distance of the node from its neighbors is large, we give low points to the node. In contrast, if the mean distance of the node from its neighbors is small, we give high points to the node. Therefore, the density of the node reflects the number of nodes close to it and also the number of samples assigned to it. We take the density of the node as a measurement of the importance of the samples assigned to the node. After the LB-SOINN finishes training on the samples from the current task, we sort the samples according to the density of the node they are assigned to and the top K samples are selected to write to the memory. We divide the memory equally for each of the previous tasks, so $K = M/t$, where M is the total memory size and t is the number of observed tasks, including the current task. The old memory consists of samples from the previous $t-1$ tasks and each task keeps $M/(t-1)$ samples in the old memory. For each of the $t-1$ tasks, the $M/(t-1) - M/t$ samples with the lowest node densities are deleted, resulting in K empty slots in the memory, which is then rewritten by the selected K samples in the current task.

5 Experiments

5.1 Experimental Settings

For each task, we randomly sample 5000 tweets from the 80% of the collected data for training, 10% of the collected data for testing, and the rest 10% for development. We allow the model to make more than one pass over the training samples in the current task or the current memory during training. We use average macro F1 score and average micro F1 score for evaluation.

$$\text{Average F1:} AvgF1(t) = \frac{1}{t}\sum_{i=1}^{t} F1_{t,i} \quad (7)$$

where $F1_{t,i}$ is the F1 score, either macro F1 or micro F1, achieved by the model on the ith task after being trained on the tth task. The larger this metric, the better the model. We compare our methods with the following methods:

Fine-tuning: The model contains two bidirectional LSTM encoders (Hochreiter and Schmidhuber, 1997; Zhou et al., 2016; Liu et al., 2016) to encode the tweet and the group separately. The score of the group is calculated as the cosine distance between the hidden state of the tweet encoder and that of the group encoder. This model is also the backbone model of all the methods described below, except Fine-tuning + BERT. The model is directly fine-tuned on the stream of tasks, one after another, by the ranking loss function in 1.

Fine-tuning+BERT: The training framework is the same as above, but each encoder is replaced by a pre-trained BERT model (Devlin et al., 2019) followed by a linear layer. The linear layers are fine-tuned during training.

Fine-tuning+RMR (Random Memory Replay): We augment the fine-tuning method with an additional memory module. Same as in section 4.2, the memory is divided equally for each task, but instead of using LB-SOINN, the K samples are randomly sampled from the current training data and then rewrite K random slots in the old memory.

EWC: EWC is a regularization-based method, adding a penalty term $\sum_i \frac{\lambda}{2} F_i(\theta_i - \theta_i^*)^2$ to the ranking loss function 1. F_i is the diagonal of the Fisher information matrix F, θ is the model parameter, and i labels each parameter. θ^* is the model parameter when the model finishes training on the previous task. λ is set to 2e6 in our experiments.

GEM: We use the episodic memory in the original paper: the memory is populated with m random

Number of observed tasks	t=5		t=10		t=15	
Avg F1 score (%)	Macro	Micro	Macro	Micro	Macro	Micro
Multitask	15.26	67.07	5.05	37.20	3.57	38.61
Fine-tuning	6.02	16.44	4.35	5.77	3.96	6.18
Fine-tuning + BERT	6.02	16.44	4.06	5.45	3.03	5.80
Fine-tuning + RMR	11.15	44.40	2.56	15.77	3.51	15.19
EWC	8.57	20.42	2.42	6.81	1.95	7.27
GEM	**13.04**	30.95	3.07	12.51	2.70	15.07
Ours	12.61	**49.75**	**6.96**	**47.30**	**5.13**	**44.62**

Table 2: Experimental results. RMR: random memory replay. The best results are in bold.

Number of observed tasks	t=5		t=10		t=15	
Avg F1 score (%)	Macro	Micro	Macro	Micro	Macro	Micro
Full Model	12.61	49.75	**6.96**	**47.30**	5.13	**44.62**
w/o D_{KLmem}	**15.00**	**58.64**	4.21	36.36	3.72	40.87
w/o VRL	11.05	35.03	4.53	13.69	3.65	11.28
w/o LB-SOINN	13.01	50.99	6.15	44.42	**5.59**	30.91

Table 3: Ablation study. w/o D_{KLmem}: D_{KLmem} in the equation 6 is removed. w/o VRL: VRL is replaced by the model used in the fine-tuning setting, i.e., fine-tuning + LB-SOINN memory replay. w/o LB-SOINN: LB-SOINN memory replay is replaced by random memory replay, i.e., VRL + RMR. The best results are in bold.

samples from each task. m is a predefined size of the episodic memory. We set $m = 100$ in our experiments, so each task can add 100 tweets to the memory. By the end of the 15 tasks, the total memory of GEM contains 1500 tweets.

Multitask Learning: The tasks are trained simultaneously. We mix the training data from multiple tasks to train the model. This setting does not follow the lifelong learning setting where the tasks are trained sequentially. We add this setting in our experiments to show the potential room for improvement concerning each lifelong learning method.

We do not compare our method with Support Vector Machine (Suykens and Vandewalle, 1999) or Logistic Regression, because they require the number of classes to be fixed and to be known in advance, which is unrealistic in our tasks. We also do not compare our method with Qian et al. (2018a) since the latter also has this requirement, as mentioned in section 3. Adapting their method for the lifelong learning setting requires modifying both the model architecture and the training algorithm, which is beyond the scope of this paper.

In all our experiments, we use 1-layer bi-LSTM as encoders except the fine-tuning + BERT setting and we use cosine distance to measure similarity. The input of the group encoder is the concatenation of the group name and its hate ideology. We use 1-layer bidirectional GRU (Cho et al., 2014) as the decoder in VRL. The hidden size of the encoders and the decoders is 64. The latent variable size in VRL is 128. We use 300-dimensional randomly initialized word embeddings. All the neural networks are optimized by Adam optimizer with the learning rate 1e-4. The batch size is 64. The loss margin $m = 0.5$. The maximum number of training epochs for each task is set to 20. For LB-SOINN, $\lambda = 1000$, $\eta = 1.04$. The memory size is limited to 1000 tweets for all the methods using a memory module except GEM. We do not set episodic memory size for each task as GEM because for lifelong hate speech classification, the number of tasks keeps increasing in the real world, and assuming unlimited total memory is unrealistic.

5.2 Experimental Results

The experimental results are shown in Table 2. We report the performance of each method after the model finishes training on the first 5 tasks, first 10 tasks, and all the 15 tasks. The average macro-F1 score is much lower than the average micro-F1 score due to the imbalanced data of each task. The large performance gap between the multitask training and fine-tuning shows that there exists severe catastrophic forgetting and that the low average F1 scores in the fine-tuning setting are not due to the model capacity. Replacing the bi-LSTM encoder with the pre-trained BERT encoder does not improve the performance.This reconfirms that the low scores result from catastrophic forgetting, not model capacity. Actually fine-tuning and fine-tuning with BERT achieves the same average F1 scores at $t = 5$ because both models completely forget the previous tasks after converging on the fifth

task, so both models achieve the same F1 scores on the testing data of the fifth task while achieving 0 scores on the previous four tasks. Due to the large model capacity of BERT, fine-tuning with BERT tends to overfit on the training data more seriously, leading to slight performance decline at $t = 10$ and $t = 15$ compared to using bi-LSTM encoders. Since model capacity is not the key factor to solve catastrophic forgetting, we simply use bi-LSTM as encoders in our model instead of BERT, considering the computational cost.

Adding RMR to the fine-tuning setting achieves significant performance improvement, even better than EWC or GEM. This is related to the characteristic of our tasks mentioned at the end of section 3. EWC remembers previous tasks by slowing down the update of the model parameters important to them, which is more suitable for the sequence of tasks that are similar to each other. However, significant changes in vocabulary, topic, or input data distribution are very common in our sequence of tasks, making memory replay more efficient than EWC. The performance of GEM during the second half of the training is close to that of fine-tuning with RMR, but there exists a gap in the first half. The reason is that GEM sets an episodic memory for each task, of which the size is 100 in our experiments, so before the 10th task in the sequence, the size of the total memory available for GEM is less than that of the memory module used in the fine-tuning with RMR setting.

Although RMR improves the performance, the average F1 scores still drop quickly when the number of tasks increases. In the late stage of sequential training, each task can only keep dozens of samples in the memory and the model is not able to generalize well based on the memory. Our method solves this problem by combining VRL and LB-SOINN memory replay. The performance of our model is better and more stable than the other methods when the number of tasks increases. Our method achieves higher scores than multitask training in the last four columns of Table 3 because learning on one task is easier than learning on a mix of tasks simultaneously. Every model in our sequential training experiments can easily achieve high F1 scores on the current task, making a large contribution to the average F1 scores. However, when doing multitask training, the model loses this benefit.

To investigate the effect of our method, we conduct the ablation study as shown in Table 3. Removing D_{KLmem} from the final loss function in equation 6 does not lower the performance when the number of observed tasks is small ($t=5$) because each task can store a few hundreds of samples in the memory at the early stage of sequential training, which is sufficient for the model to learn the previous tasks. However, when the number of tasks increases, D_{KLmem} shows its effect on alleviating catastrophic forgetting.

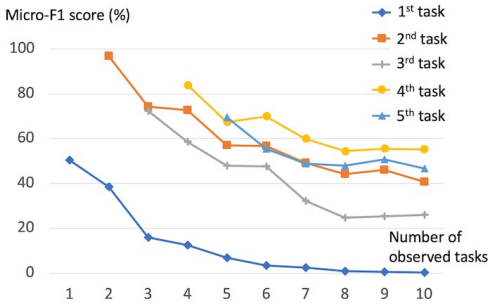

Figure 3: The testing results of the first 5 tasks in the sequence when our model is trained on the first 10 tasks.

Fine-tuning+LB-SOINN (Table 3) does not perform as well as fine-tuning+RMR (Table 2), while VRL+LB-SOINN (i.e., full model) performs better than VRL+RMR (Table 3). The reason lies in the input for LB-SOINN.Compared to the hidden representations spread evenly in the hidden space, the hidden representations which are well-organized in different group clusters make it easier for LB-SOINN to learn a reasonable topology structure of the training samples. VRL achieves this by explicitly pushing the hidden representation of tweets to follow a learned multivariate Gaussian distribution unique to each group. On the other hand, directly using the hidden state of the tweet encoder does not exhibit such kind of characteristics. VRL not only distills task knowledge but also provides an appropriate input for LB-SOINN, as stated in section 4.

5.3 Error Analysis

Although our model achieves significant improvement over the baseline methods, we observe that our method does not perform well on the first task. As shown in Figure 3, there exists a large gap between the performance on the first task and the other tasks, and the micro-F1 score on the first task quickly drops to almost 0 when the number of observed tasks increases. We find the same results after we change the order of tasks in the sequence, so this is not the result of the task difficulty but is

the result of our method. We find this problem is due to the reconstruction loss, which is the first part in equation 4. The model observes a very limited number of tweets when training on the first task, making it difficult to learn the language model and reconstruct the tweet. As a result, the tweet representation learned on the first task may not contain the information we require, resulting in a large performance gap. When the number of observed tasks increases, this problem goes away quickly. We anticipate pre-training the VAE in our model (the left branch in Figure 2) on a large Twitter corpus can alleviate this problem at the beginning of training.

6 Conclusion

In this paper, we introduce the lifelong hate speech classification task and propose to use the VRL and LB-SOINN memory module to alleviate catastrophic forgetting. Our proposed method has the potential to benefit other lifelong learning tasks where the similarity between the contiguous tasks can be low. We intend to make our implementation freely available to facilitate more application and investigation of our method in the future.

References

Pinkesh Badjatiya, Shashank Gupta, Manish Gupta, and Vasudeva Varma. 2017. Deep learning for hate speech detection in tweets. In *Proceedings of the 26th International Conference on World Wide Web Companion*, pages 759–760. International World Wide Web Conferences Steering Committee.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

Pew Research Center. 2017. Online harassment 2017.

Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. 2019. Efficient lifelong learning with A-GEM. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1724–1734.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. *arXiv preprint arXiv:1703.04009*.

Cyprien de Masson d'Autume, Sebastian Ruder, Lingpeng Kong, and Dani Yogatama. 2019. Episodic memory in lifelong language learning. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 13122–13131.

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

Chrisantha Fernando, Dylan Banarse, Charles Blundell, Yori Zwols, David Ha, Andrei A. Rusu, Alexander Pritzel, and Daan Wierstra. 2017. Pathnet: Evolution channels gradient descent in super neural networks. *CoRR*, abs/1701.08734.

Robert M French. 1999. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135.

Lei Gao, Alexis Kuppersmith, and Ruihong Huang. 2017. Recognizing explicit and implicit hate speech using a weakly supervised two-path bootstrapping approach. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 774–782.

Jennifer Golbeck, Zahra Ashktorab, Rashad O Banjo, Alexandra Berlinger, Siddharth Bhagwan, Cody Buntain, Paul Cheakalos, Alicia A Geller, Quint Gergory, Rajesh Kumar Gnanasekaran, et al. 2017. A large labeled corpus for online harassment research. In *Proceedings of the 2017 ACM on Web Science Conference*, pages 229–233. ACM.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526.

Xialei Liu, Marc Masana, Luis Herranz, Joost Van de Weijer, Antonio M Lopez, and Andrew D Bagdanov. 2018. Rotate your networks: Better weight consolidation and less catastrophic forgetting. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 2262–2268. IEEE.

Yang Liu, Chengjie Sun, Lei Lin, and Xiaolong Wang. 2016. Learning natural language inference using bidirectional lstm model and inner-attention. *arXiv preprint arXiv:1605.09090*.

David Lopez-Paz and Marc'Aurelio Ranzato. 2017. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems*, pages 6467–6476.

James L McClelland, Bruce L McNaughton, and Randall C O'Reilly. 1995. Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory. *Psychological review*, 102(3):419.

Michael McCloskey and Neal J Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier.

Chikashi Nobata, Joel Tetreault, Achint Thomas, Yashar Mehdad, and Yi Chang. 2016. Abusive language detection in online user content. In *Proceedings of the 25th international conference on world wide web*, pages 145–153. International World Wide Web Conferences Steering Committee.

Jose L Part and Oliver Lemon. 2016. Incremental online learning of object classes using a combination of self-organizing incremental neural networks and deep convolutional neural networks. In *Workshop on Bio-inspired Social Robot Learning in Home Scenarios (IROS), Daejeon, Korea*.

Jose L Part and Oliver Lemon. 2017. Incremental online learning of objects for robots operating in real environments. In *2017 Joint IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*, pages 304–310. IEEE.

Jing Qian, Mai ElSherief, Elizabeth Belding, and William Yang Wang. 2018a. Hierarchical cvae for fine-grained hate speech classification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3550–3559.

Jing Qian, Mai ElSherief, Elizabeth Belding, and William Yang Wang. 2018b. Leveraging intra-user and inter-user representation learning for automated hate speech detection. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 118–123.

Roger Ratcliff. 1990. Connectionist models of recognition memory: constraints imposed by learning and forgetting functions. *Psychological review*, 97(2):285.

Hippolyt Ritter, Aleksandar Botev, and David Barber. 2018. Online structured laplace approximations for overcoming catastrophic forgetting. In *Advances in Neural Information Processing Systems*, pages 3738–3748.

Andrei A. Rusu, Neil C. Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. 2016. Progressive neural networks. *CoRR*, abs/1606.04671.

Fan-Keng Sun, Cheng-Hao Ho, and Hung-Yi Lee. 2020. LAMOL: language modeling for lifelong language learning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*.

Johan AK Suykens and Joos Vandewalle. 1999. Least squares support vector machine classifiers. *Neural processing letters*, 9(3):293–300.

Hong Wang, Wenhan Xiong, Mo Yu, Xiaoxiao Guo, Shiyu Chang, and William Yang Wang. 2019. Sentence embedding alignment for lifelong relation extraction. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 796–806.

William Warner and Julia Hirschberg. 2012. Detecting Hate Speech on the World Wide Web. In *ACL'12: Proceedings of the 2nd Workshop on Language in Social Media*, pages 19–26. Association for Computational Linguistics.

Zeerak Waseem and Dirk Hovy. 2016. Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In *Proceedings of the NAACL student research workshop*, pages 88–93.

Rui Xia, Jie Jiang, and Huihui He. 2017. Distantly supervised lifelong learning for large-scale social media sentiment analysis. *IEEE Transactions on Affective Computing*, 8(4):480–491.

Friedemann Zenke, Ben Poole, and Surya Ganguli. 2017. Continual learning through synaptic intelligence. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 3987–3995.

Hongwei Zhang, Xiong Xiao, and Osamu Hasegawa. 2013. A load-balancing self-organizing incremental neural network. *IEEE Transactions on Neural Networks and Learning Systems*, 25(6):1096–1105.

Haoti Zhong, Hao Li, Anna Cinzia Squicciarini, Sarah Michele Rajtmajer, Christopher Griffin, David J Miller, and Cornelia Caragea. 2016. Content-Driven Detection of Cyberbullying on the Instagram Social Network. In *IJCAI'16: Proceedings of the 25th International Joint Conference on Artificial Intelligence*, pages 3952–3958.

Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. 2016. Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Short Papers)*, volume 2, pages 207–212.