Merging Statistical Feature via Adaptive Gate for Improved Text Classification

Xianming Li, [1†] **Zongxi Li,** [2† *] **Haoran Xie,** [3] **Qing Li** [4]

[1] Ant Group, Shanghai, China
[2] Department of Computer Science, City University of Hong Kong, Hong Kong SAR
[3] Department of Computing and Decision Sciences, Lingnan University, Hong Kong SAR
[4] Department of Computing, The Hong Kong Polytechnic University, Hong Kong SAR
niming.lxm@antgroup.com, zongxili2-c@my.cityu.edu.hk, hrxie@ln.edu.hk, csqli@comp.polyu.edu.hk

Abstract

Currently, text classification studies mainly focus on training classifiers by using textual input only, or enhancing semantic features by introducing external knowledge (e.g., hand-craft lexicons and domain knowledge). In contrast, some intrinsic statistical features of the corpus, like word frequency and distribution over labels, are not well exploited. Compared with external knowledge, the statistical features are deterministic and naturally compatible with corresponding tasks. In this paper, we propose an Adaptive Gate Network (**AGN**) to consolidate semantic representation with statistical features selectively. In particular, AGN encodes statistical features through a variational component and merges information via a well-designed valve mechanism. The valve adapts the information flow into the classifier according to the confidence of semantic features in decision making, which can facilitate training a robust classifier and can address the overfitting caused by using statistical features. Extensive experiments on datasets of various scales show that, by incorporating statistical information, AGN can improve the classification performance of CNN, RNN, Transformer, and Bert based models effectively. The experiments also indicate the robustness of AGN against adversarial attacks of manipulating statistical information.

1 Introduction

Text classification is playing an essential role in Natural Language Processing (NLP) as one of the fundamental tasks with broad applications. In recent years, CNN-based (Kim 2014; Lai et al. 2015; Conneau et al. 2017; Johnson and Zhang 2017), RNN-based (Socher et al. 2013; Graves, Jaitly, and Mohamed 2013; Tang, Qin, and Liu 2015), Transformer-based (Vaswani et al. 2017), and Bert-based (Devlin et al. 2019; Lan et al. 2020) deep learning models have become mainstream approaches which outperform traditional classification methods. To enrich semantic features, researchers turn to some external knowledge, such as character, sentiment lexicon, and entity knowledge base, as complementary information (Post and Bergsma 2013; Teng, Vo, and Zhang 2016; Chen et al. 2019). These studies show that introducing proper external knowledge is helpful to the classification task. However, we notice that the current deep learning paradigm has overlooked such primitive features as word frequency and distribution, which are fixed, intrinsic, and easy-to-retrieve features of a corpus (Yang and Pedersen 1997; Aizawa 2003). The most representative algorithm utilizing statistical features is still the *term frequency-inverse document frequency* (TFIDF), a straightforward information retrieval technique for document modelling. However, because of the bag-of-word nature, TFIDF is unable to utilize positional information and capture the fine-grained semantics (Ramos et al. 2003), which makes it less favourable compared with other representation learning methods in the deep architecture. From our pilot study, we find that using statistical features (such as term-count-of-label to be defined in Section 3.1) as an additional feature brings forth substantial improvements to various baselines, in which the word frequency adapts weights of terms via an attention layer. Unfortunately, earlier research may have underestimated the real power of corpus-level statistic features in deep learning, and new fusion mechanism is necessary to incorporate such information. In particular, when designing the fusion mechanism, we must consider two major concerns:

1. The semantic feature and statistical feature are not compatible in scale and dimension;
2. The new information may not be necessary for all semantic features.

In this paper, we advocate a new framework, Adaptive Gate Network (AGN)[1], to enhance neural classification by fusing statistical features elegantly via a gate mechanism. More concretely, the AGN consists of three components, a *variational encoding network* referred to as the V-Net, a *semantic representation projection network* referred to as the S-Net, and an *adaptive gate mechanism* denoted as the valve. The V-Net exploits unsupervised autoencoder to learn a global representation for each statistical feature vector, where we notice that employing a variational inference can further improve the model performance compared with a vanilla autoencoder. The S-Net extracts latent semantic representation from textual input by using one of the most commonly used extractors, i.e., CNN, RNN, Transformer, or Bert. Furthermore, the S-Net projects semantic features into

*Corresponding author; † Equal contribution.

[1] Code available at https://github.com/4AI/AGN

an information space via sigmoid activation, where the value of each neuron indicates the confidence of corresponding semantic feature in decision-making. Intuitively, a feature is high-confident and decisive if its value is either near 0 or 1 after activation, and a feature is low-confident if its activated value is around 0.5. The valve component aligns information from two sources and adapts the information flow. To address the first concern listed above, we employ a nonlinear projection to map statistical features into the shared information space, making both latent representations compatible with each other. The second concern raises a new perspective on the method of using *additional information*[2]. Our standpoint is that not all semantic features need to be enhanced since some may introduce noise to the classifier. Therefore, instead of an element-wise operation, the valve module adds auxiliary information to the less-confident semantic features while the high-confident semantic features remain unchanged. By doing this, the proposed AGN model can achieve better decision making by balancing between the original semantic features and the additional features.

Besides, considering that utilizing statistical features tend to have undesired bias information due to limited data size, we exploit statistics from a large dataset to a small dataset to explore how the statistical bias is affected by the data amount and how such a deviation compromises the model performance. Furthermore, to address the issue that statistical features are easy to be attacked, we conduct adversarial attacks by manipulating the statistical information to demonstrate the robustness of our proposed AGN, which aims to validate that the valve component can effectively filter out corrupted information. The main contributions of this paper are summarized as follows:

- To the best of our knowledge, we are the first to leverage corpus-level statistics explicitly in deep architecture for text classification and prove it as an effective approach.
- To fuse statistics feature into semantic features with low confidence, we propose a novel Adaptive Gate Network to retrieve necessary and useful global information.
- We conduct extensive experiments on seven datasets of different scales and topics. The results show that our proposed model produces significant improvements on baseline models. Furthermore, we conduct additional experiments to demonstrate the robustness of the proposed AGN against biased statistics and adversarial attacks.

2 Related Work

2.1 Text classification

Existing approaches employing deep architecture for supervised classification have achieved much remarkable progress. Kim (2014) proposed a classic TextCNN model to extract local and position-invariant features. Zhang, Zhao, and LeCun (2015) applied CNN to model character-level features and achieve competitive performance. Hochreiter and Schmidhuber (1997) and Socher et al. (2013) used recursive networks explicitly exploiting time-series features, based on which several variants of the recurrent model are proposed, including BiLSTM (Graves, Jaitly, and Mohamed 2013) and GRU (Chung et al. 2014) with more complex gate mechanisms. However, these methods may not give enough weights to some discriminative words. To address this problem, Bahdanau, Cho, and Bengio (2015) introduced and applied attention mechanism to machine translation. Afterwards, the attention mechanism has been widely applied in various NLP tasks. Yang et al. (2016) proposed the Hierarchical Attention Network to imitate the hierarchical structure of sentences and capture both word-level and sentence-level features. Vaswani et al. (2017) stacked multiple blocks of self-attention to produce a more robust sentence-level representation by learning global dependency. Devlin et al. (2019) combined Transformer-based architecture and a large corpus to maximize Transformer's ability. Tang et al. (2020) combined a graph-based model with Transformer blocks to enhance sentiment classification. These works mainly focus on architecture design for feature extraction. In contrast, we propose to merge additional information through an adaptive fusion mechanism.

2.2 Classifier with additional knowledge

There have been numerous works on exploiting external knowledge in NLP. Researchers have created and exploited many active features incorporating information from various domains, including but not limited to linguistics, psychology and knowledge base. Post and Bergsma (2013) utilized syntactic structure features such as POS tagging and dependency parsing to improve classification performance. Teng, Vo, and Zhang (2016), Liang et al. (2018), and Rojas et al. (2020) fused emotional lexicon into the model framework for sentiment analysis. Wang et al. (2017) conceptualized sentence as a set of concepts using the taxonomy knowledge base, and obtained the embeddings by merging concepts on top of pretrained word vectors, so as to capture ampler contextual information facilitated by deep models. Chen et al. (2019) introduced conceptual information and entity links from knowledge base into the model pipeline via attention mechanism. These works, however, pay little attention to the necessity and compatibility of the added information, thus cannot avoid bringing noise to the classifier.

3 Methodology

The overall framework of our proposed approach is shown in Figure 1, and in this section, we introduce our framework in terms of its components.

3.1 Global information

We first give a formal definition of *term-count-of-labels* as statistics of terms towards labels.

Definition 1 Given a word w and a set of labels of c classes, the term-count-of-labels (TCoL) vector of w is

$$\boldsymbol{\zeta}^w = [\zeta_1, \ldots, \zeta_c], \tag{1}$$

[2]For concise expression, we use *additional information* to denote both statistical feature and external knowledge, which are additional to the semantic features, in the remaining part of this paper

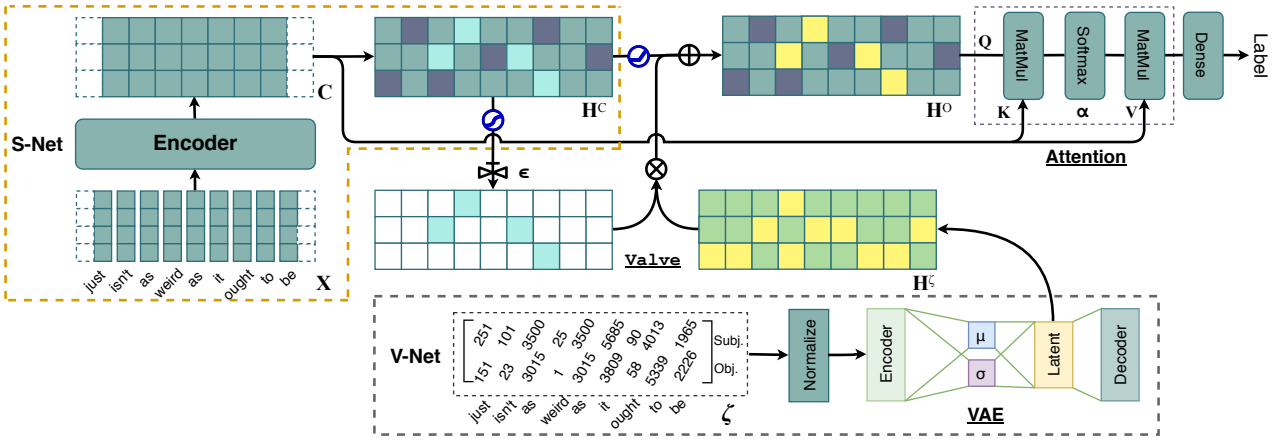


Figure 1: The generic framework of the proposed AGN. The *subj.* and *obj.* are labels of the Subj dataset.

where ζ_i is the count of word w on label i. Given a sentence $s = \{w_i\}_{i=1}^m$, the TCoL matrix of sentence s is

$$\boldsymbol{\zeta}^s = [\boldsymbol{\zeta}^{w_1}, \ldots, \boldsymbol{\zeta}^{w_m}]. \quad (2)$$

The notion of TCoL is to capture the global distribution of labels as features of a word. Such features are primitive but highly informative for feature selection and information retrieval by determining word relevance (Salton and Buckley 1988; Ramos et al. 2003). Intuitively, if a word w has very high or very low frequency on all labels, then we can assume that w has a limited contribution to the classification task. In contrast, if a word appears more frequently in specific label class, we assume this word is discriminative. Note that the TCoL dictionary V is obtained from the training set only. An extensive discussion on the effect of TCoL on the model performance is given in Section 5.2.

3.2 V-Net: Variational Encoding Network

The goal of V-Net (of Figure 1) is to transform statistical features into effective representations. TCoL consists of integer counts of terms, which is initially not compatible with semantic features in both dimension and scale. V-Net employs an autoencoder to map the discrete TCoL vectors into a latent continuous space to obtain the global representation of statistical information. Moreover, we notice that the representation encoded by bounding the latent space with a multivariate Gaussian distribution can produce substantial improvements to the classifier compared with that encoded by a vanilla autoencoder. Therefore, in this work, we adopt the Variational Autoencoder (VAE) (Kingma and Welling 2014) to encode the TCoL.

We generate TCoL for all sentences in a dataset and obtain $\mathbf{Z} = \{\boldsymbol{\zeta}^s_{(i)}\}_{i=1}^N$, which consists of N i.i.d. discrete TCoL variable $\boldsymbol{\zeta}$. We assume all TCoL vectors are generated by a random process $p_{\boldsymbol{\theta}}(\boldsymbol{\zeta}|\mathbf{z})$, involving a latent variable $\mathbf{z}$ which is sampled from a prior distribution $p_{\boldsymbol{\theta}}(\mathbf{z})$. Since the posterior $p_{\boldsymbol{\theta}}(\mathbf{z}|\boldsymbol{\zeta})$ is intractable, we cannot directly learn the generative model parameters $\boldsymbol{\theta}$. Thus, we adopt the variational approximation $q_{\boldsymbol{\phi}}(\mathbf{z}|\boldsymbol{\zeta})$ to learn the variational parameters $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$ jointly. Consequently, we can optimize the model by maximizing the marginal likelihood that is composed of a sum over the marginal likelihoods of individual $\boldsymbol{\zeta}$:

$$\log p_{\boldsymbol{\theta}}(\boldsymbol{\zeta}) = D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\boldsymbol{\zeta}) \| p_{\boldsymbol{\theta}}(\mathbf{z}|\boldsymbol{\zeta})) + \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{\zeta}). \quad (3)$$

Since the KL divergence term is non-negative, we can derive the likelihood term $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{\zeta})$ to obtain the variational lower bound on the marginal likelihood, i.e.,:

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{\zeta}) = -D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\boldsymbol{\zeta}) \| p_{\boldsymbol{\theta}}(\mathbf{z})) + \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\boldsymbol{\zeta})} \left[\log p_{\boldsymbol{\theta}}(\boldsymbol{\zeta}|\mathbf{z})\right], \quad (4)$$

where the KL term has a closed-form solution, and the expectation term is the reconstruction error. We adopt the reparameterization trick to fit the variational framework into an autoencoder. We employ two encoders to generate two sets of μ and σ as the values of prior distribution's mean and standard deviation, respectively. Since our approximate prior is a multivariate Gaussian, we denote the variational posterior with a diagonal covariance structure:

$$\log q_{\boldsymbol{\phi}}(\mathbf{z}|\boldsymbol{\zeta}) = \log \mathcal{N}(\mathbf{z}; \mu, \sigma^2 \mathbf{I}). \quad (5)$$

By training the unsupervised VAE model, we can obtain the latent variables $\boldsymbol{\zeta}^{\mathbf{z}}$ via the probabilistic encoder, which will be the global representation of TCoL. The training of V-Net is independent of the main classifier, and the representation $\boldsymbol{\zeta}^{\mathbf{z}}$ is generated during the preprocessing stage and will be fed into the classifier via the valve component.

3.3 S-Net: Semantic Representation Projection Network

The function of S-Net (of Figure 1) is to extract semantic features from textual input and project semantic features into the information space for confidence evaluation. The input of S-Net is the sentence s with a fixed length m. For non-Bert models, we first map each word into a k-dimensional continuous space and obtain the word embedding vector $\mathbf{x}_i \in \mathbb{R}^k$. Then we concatenate all word vectors to form a $k \times m$ matrix as the model input: $\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m]$. We pad the sentences to maintain a uniform length for all sentences. Then we apply semantic feature extractor (i.e., CNN,

LSTM, and Transformer) on the embedding matrix to produce latent semantic feature map $\mathbf{C}$:

$$\mathbf{C} = \text{FeatureExtractor}(\mathbf{x}). \tag{6}$$

As for Bert model, we extract feature map via the pretrained Bert base over text input:

$$\mathbf{C} = \text{Bert}(s). \tag{7}$$

Then we map the semantic feature map $\mathbf{C}$ into an information space through a dense layer:

$$\mathbf{H}^C = \mathbf{W}^C \cdot \mathbf{C} + \mathbf{b}^C, \tag{8}$$

The values in the sigmoid-activated representation, $\mathbf{H'}^C = \sigma(\mathbf{H}^C)$, where $\sigma(\cdot)$ is the sigmoid function, are exploited to evaluate the confidence of corresponding semantic features in the decision-making process.

3.4 Valve Component

As stated in Section 3.2, the representation $\boldsymbol{\zeta}^{\mathbf{z}}$ of TCoL is obtained offline. In order to exploit statistical features flexibly, we apply a dense layer to project $\boldsymbol{\zeta}^{\mathbf{z}}$ into the information space that is shared with semantic features:

$$\mathbf{H}^{\zeta} = \mathbf{W}^{\zeta} \cdot (\boldsymbol{\zeta}^{\mathbf{z}}) + \mathbf{b}^{\zeta}. \tag{9}$$

The valve component fuses $\mathbf{H}^C$ and $\mathbf{H}^{\zeta}$ to output a statistical information-enhanced semantic feature map $\mathbf{H}^O$ through the $\mathbf{AdaGate}$ function,

$$\begin{aligned} \mathbf{H}^O &= \mathbf{AdaGate}(\mathbf{H}^C, \mathbf{H'}^C, \mathbf{H}^{\zeta}, \epsilon) \\ &= \text{ReLU}(\mathbf{H}^C) + \mathbf{Valve}(\mathbf{H'}^C, \epsilon) \odot \mathbf{H}^{\zeta}, \end{aligned} \tag{10}$$

where $\text{ReLU}(\cdot)$ is the activation function, and $\odot$ stands for an element-wise product. The values in $\mathbf{H'}^C$ are in probability form, and the $\mathbf{Valve}$ function is designed to restore less-confident entries (with probability near 0.5) for matching with elements in $\mathbf{H}^{\zeta}$. Concretely, for every unit $a \in \mathbf{H'}^C$,

$$\mathbf{Valve}(a, \epsilon) = \begin{cases} a, & \text{if } 0.5 - \epsilon \leq a \leq 0.5 + \epsilon \\ 0, & \text{otherwise} \end{cases} \tag{11}$$

where ϵ is a leaky hyper-parameter tuning the threshold of confidence. Specifically, we dump all statistical information if $\epsilon = 0$, and accept all statistical information if $\epsilon = 0.5$. Therefore, the element-wise production exploits $\mathbf{Valve}(\cdot, \epsilon)$ as a filter to extract necessary information only.

3.5 Classifier

We employ attention to combine the consolidated semantic representation $\mathbf{H}^O$ with the original feature map $\mathbf{C}$:

$$Attention(\mathbf{H}^O, \mathbf{C}) = \text{softmax}(\mathbf{H}^O \mathbf{C}^{\intercal})\mathbf{C}. \tag{12}$$

Note that if we reject all statistical information (i.e., $\epsilon = 0$), Eqn. (12) will become self-attention (Vaswani et al. 2017) as $\mathbf{H}^O = \mathbf{C}$.

After passing through fully-connected layers and a softmax layer, feature vectors are mapped to the label space for label prediction and loss calculation. To maximize the probability of the correct label Y_{True}, we deploy an optimizer to minimize cross-entropy loss L,

$$L = \text{CrossEntropy}(Y_{\text{True}}, Y_{\text{Pred.}}). \tag{13}$$

4 Experiment

4.1 Datasets

We test the proposed model on the following datasets (with summary statistics in Table 2).

Subj[3] (Pang and Lee 2004) is a dataset of subjectivity. Each sentence is annotated as subjective or objective.

SST-1[4] (Socher et al. 2013) is the Stanford Sentiment Treebank dataset of reviews with five fine-grained sentiment labels (very positive/negative, positive/negative, neutral).

SST-2 (Socher et al. 2013) is the Stanford Sentiment Treebank with binary sentiment labels.

TREC[5] (Li and Roth 2002) is a question dataset with questions about the person, location, numbers, etc.

AG's News[6] (Zhang, Zhao, and LeCun 2015) consists of news articles from the AG's corpus of news articles on the web pertaining to the four largest classes.

Yelp Review Full (Yelp F.)[7] is the Yelp Open Dataset consists of reviews with polarity labels ranging from 1 to 5.

Yelp Review Polarity (Yelp P.) is the reviews subset of Yelp Open Dataset. Compared with Yelp F., Yelp P. only has binary labels (negative and positive).

We deploy 10-fold cross-validation on the dataset without standard train/test split (i.e., Subj). For datasets with standard split, we run ten trials and report the average results.

4.2 Baselines

Since our goal is to demonstrate that the consolidated semantic representation is more conducive for classification, we compare the models with and without additional knowledge using the following popular feature extractors:

TextCNN (Kim 2014): a popular CNN-based classifier exploiting one-dimensional convolution operation and max-over-time pooling.

BiLSTM (Graves, Jaitly, and Mohamed 2013): a bi-directional LSTM model extracting both forward and reverse sequential features.

Transformer (Vaswani et al. 2017) employs stacked self-attention blocks to learn semantic dependency. We use the encoder part of the Transformer followed by a classifier.

Bert (Devlin et al. 2019): the state-of-the-art framework in many NLP tasks. We fine-tuned the pretrained Bert base with classifier.

Since we adopt attention mechanism over extracted semantic feature representation, we compare our proposed AGN model against **TextCNN**+Self-Attn, **BiLSTM**+Self-Attn, **Transformer**+Self-Attn, and **Bert**+Self-Attn, where self-attention blocks are employed on the latent semantic feature map $\mathbf{C}$ obtained by Eqn. 6 and Eqn. 7.

4.3 Word embedding and parameter settings

To focus on the effect of statistical features and AGN model, we randomly initialize word embedding vectors (except Bert) to eliminate the influence of different pretrained

[3] http://www.cs.cornell.edu/people/pabo/movie-review-data/

[4] http://nlp.stanford.edu/sentiment/

[5] https://trec.nist.gov/data.html

[6] http://groups.di.unipi.it/ gulli/AG_corpus_of_news_articles.html

[7] https://www.yelp.com/dataset/

Model	Accu. (%)						
	Subj	SST-1	SST-2	TREC	AGNews	Yelp F.	Yelp P.
CNN	92.61 ±0.23	43.88 ±1.09	80.13 ±0.70	90.00 ±0.59	92.10 ±0.07	94.53 ±0.26	66.98 ±0.17
CNN+S/A	92.81 ±0.62	44.08 ±0.81	80.79 ±0.60	89.84 ±0.65	92.09 ±0.39	94.84 ±0.07	66.97 ±0.23
CNN+AGN	**93.75**§±0.68	**45.91**§±1.00	**82.24**§±0.60	**92.11**§±0.62	**92.76**‡±0.33	**95.15**‡±0.13	**67.93**§±0.13
LSTM	92.50 ±0.25	44.22 ±1.08	81.18 ±0.52	88.04 ±1.03	92.14 ±0.12	94.80 ±0.13	66.91 ±0.26
LSTM+S/A	92.57 ±0.65	44.38 ±0.74	81.13 ±0.54	88.09 ±0.98	92.24 ±0.13	94.86 ±0.21	66.96 ±0.16
LSTM+AGN	**93.39**§±0.42	**45.91**§±0.65	**82.14**§±0.57	**89.91**‡±0.74	**92.51**†±0.18	**95.05**†±0.18	**67.19**†±0.16
Trans.	87.57 ±0.46	35.62 ±0.84	67.80 ±2.62	86.70 ±0.68	89.10 ±0.14	91.89 ±0.11	63.08 ±0.12
Trans.+S/A	87.70 ±0.97	36.01 ±0.65	68.10 ±2.24	85.16 ±2.00	89.44 ±0.16	92.10 ±0.08	63.12 ±0.30
Trans.+AGN	**88.39**†±0.58	**36.78**‡±0.44	**70.32**§±1.10	**87.58**‡±1.27	**90.00**‡±0.13	**92.34**†±0.10	**63.90**‡±0.27
Bert	96.98 ±0.73	53.70 ±3.10	90.41 ±0.14	95.57 ±0.25	93.29 ±0.52	94.53 ±0.39	68.17 ±0.09
Bert+S/A	97.03 ±0.71	53.78 ±3.12	91.07 ±0.21	96.48 ±0.23	93.30 ±0.50	95.73 ±0.36	68.46 ±0.09
Bert+AGN	**97.89**§±0.71	**55.72**§±3.12	**93.27**§±0.20	**97.65**§±0.22	**93.82**†±0.47	**96.79**§±0.32	**70.74**§±0.09

$^{\dagger}p < .05$, $^{\ddagger}p < .01$, $^{\S}p < .001$.

Table 1: Results of Accuracy on all datasets. Indicated p-value means our method has significant improvement.

Data	c	l	Train	Test
Subj	2	23	$10,000$	CV
SST-1	5	18	$11,855$	$2,210$
SST-2	2	19	$9,613$	$1,821$
TREC	6	10	$5,952$	$9,125$
AG's News	4	45	$120,000$	$7,600$
Yelp F.	5	159	$650,000$	$50,000$
Yelp P.	2	153	$560,000$	$38,000$

Table 2: Summary statistics for the datasets. c: Number of classes. l: Average sentence length. *Train*: Dataset size. *Test*: Size of test set (CV means no standard train/test split).

language model. The preprocessing of textual data on all datasets follows that of Kim (2014).

The hyperparameters involved follow the settings as follows. The CNN-based models have a filter size of $[3, 4, 5]$ with 100 filters of each, and the RNN-based models have hidden dimension of 128. For the Transformer, we use an encoder with 8 heads and 3 blocks. The employed Bert model is the Bert-base Uncased, including 12 layers, 768 hidden units, and 110M parameters. We adopt Adam optimizer with a batch size of 64 for non-Bert models and 16 for Bert models. The dropout rate is set to 0.5. We conduct parameter search to test model performance with different valve size ϵ.

4.4 Experiment Results

The results of our model against other methods are listed in Table 1 (Accuracy) and Table 3 (F1 Score). The X+AGN means that the AGN model uses X as a semantic feature extractor. The X+S/A means the model employs self-attention on the feature map to extract dependency information.

In general, our proposed model consistently improves the performance of the baseline models (i.e., CNN, LSTM, Transformer, and Bert) on all datasets. The T-test in Tables 1 and 3 indicates that the improvements to the baseline models are significant. For example, the CNN+AGN model increases the accuracy by 2.11% and improves the F1 score by 2.16% on TREC compared with TextCNN. Remarkably, AGN produces substantial improvements in the accuracy of pretrained Bert model, i.e., 2.86% on SST-2, 2.26% on Yelp F., and 2.57% on Yelp P, which verifies the effectiveness of the proposed framework.

Moreover, we observe that adding self-attention module leads to compromised results on several datasets, i.e., CNN+S/A on TREC and AGNews, LSTM+S/A on SST-2, and Trans+S/A on TREC, and limited improvements on the other datasets. In contrast, the proposed AGN model yields significant improvements on all X+S/A models. This observation substantiates the effectiveness of incorporating statistical information and the proposed merging mechanism.

5 Discussion

In this section, we provide in-depth discussions regarding each component of AGN with additional experiments.

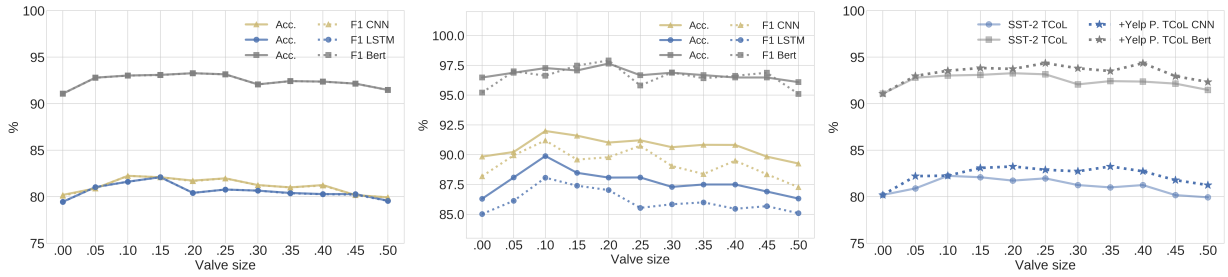
5.1 Effect of valve

We note that the improvements brought forth by TCoL can be affected by the leaky constant ϵ in the valve component, which defines a confidence interval to trigger the information fusion. To explore the effect of ϵ, we train the proposed model with different values of ϵ on two datasets, SST-2 and TREC. The results are reported in Figure 2. From the figure, we can see that the valve component is rather effective for combining knowledge from different sources. In particular, the models adaptively exploiting partial additional knowledge outperform those without additional knowledge ($\epsilon = 0$) and those with full-use of additional knowledge ($\epsilon = 0.5$). This observation supports our initial argument that the additional information is generally useful to the classifier, but not all statistical features are helpful since some may introduce noise to the classifier. These results verify the effectiveness of the valve component.

Model	F1 Score (%)						
	Subj	SST-1	SST-2	TREC	AGNews	Yelp F.	Yelp P.
CNN	92.60 ±0.22	41.79 ±0.96	80.13 ±0.70	88.59 ±1.07	92.10 ±0.06	92.96 ±0.30	58.22 ±0.35
CNN+S/A	92.80 ±0.62	41.50 ±1.01	80.78 ±0.60	89.19 ±0.65	92.06 ±0.39	93.00 ±0.10	58.63 ±0.78
CNN+AGN	**93.75**§±0.68	**42.66**§±0.87	**82.26**§±0.60	**90.75**§±0.80	**92.77**‡±0.33	**93.76**‡±0.22	**59.39**§±0.41
LSTM	92.50 ±0.25	42.51 ±0.59	81.17 ±0.53	86.21 ±0.72	92.13 ±0.12	93.59 ±0.15	57.71 ±0.13
LSTM+S/A	92.56 ±0.64	42.62 ±0.46	81.12 ±0.54	87.03 ±0.83	92.20 ±0.13	93.42 ±0.32	57.81 ±0.59
LSTM+AGN	**93.21**§±0.42	**44.42**§±0.95	**82.09**‡±0.59	**88.07**§±1.05	**92.51**†±0.18	**93.68**†±0.15	**58.50**‡±0.38
Trans.	87.55 ±0.47	31.88 ±0.96	67.61 ±2.60	85.22 ±0.81	89.00 ±0.13	89.42 ±0.16	53.11 ±0.31
Trans.+S/A	87.69 ±0.99	31.99 ±0.81	67.69 ±2.20	85.25 ±2.12	89.51 ±0.19	89.65 ±0.14	53.09 ±0.41
Trans.+AGN	**88.34**‡±0.59	**32.55**‡±0.66	**69.71**§±1.04	**86.10**‡±1.10	**89.97**†±0.13	**89.89**†±0.14	**53.97**§±0.24
Bert	96.98 ±0.73	53.49 ±1.00	90.41 ±0.15	95.13 ±0.95	93.30 ±0.52	93.04 ±0.49	61.59 ±0.18
Bert+S/A	97.03 ±0.71	52.97 ±0.12	91.07 ±0.21	95.93 ±0.23	93.32 ±0.50	94.77 ±0.48	62.24 ±0.19
Bert+AGN	**97.87**‡±0.22	**54.95**§±0.90	**93.27**§±0.20	**97.91**§±0.89	**93.79**†±0.45	**95.85**§±0.45	**63.01**§±0.20

$^{\dagger}p < .05$, $^{\ddagger}p < .01$, $^{\S}p < .001$.

Table 3: Results of *Macro* F1 Score on all datasets. Indicated p-value means our method has significant improvement.



(a) SST-2 (b) TREC (c) SST-2 + Yelp TCoL

Figure 2: (a) (b) Effect of ϵ to the model performance on datasets with different thresholds. (c) Results on SST-2 dataset using original SST-2 TCoL together with TCoL of Yelp P.

5.2 Effect of Statistical Information

The contribution of statistical feature is distinctly evident since models of X+AGN produce significant improvements compared with models of X+S/A. In Figure 3, we visualize the heatmaps of attention weights for four example sentences from Subj dataset, which the models without statistical features fail to classify but models with statistical features classify correctly. The heatmaps indicate that incorporating statistics can effectively highlight the words that are discriminative for classification by assigning higher weights. More concretely, for sentences of label *subj*, words expressing personal feelings and perspective, like "dizzying" and "I can believe", gain more weights; meanwhile, for sentences of label *obj*, words describing facts, like "something is" and "it is", are assigned with higher weights. The visualization shows that statistical information is helpful to the decision-making by properly adjusting the attention weights.

We further test the hypothesis that the deviation from real distribution of TCoL on a small-size dataset compromises the model performance. To measure the influence of deviation, we need to define or have access to the *real* distribution — an objective obviously impossible to achieve given the nature of natural language. Nevertheless, the TCoL statistics from a larger dataset annotated with the same labels can be exploited as a perfect supplementary to modify the prior knowledge of a small-scale dataset, which can approximate the real distribution better. To test the hypothesis, we train the model on a small dataset by combining original TCoL and the TCoL from a large dataset (here, we adopted SST-2 and Yelp P. as they are the only pair of datasets satisfying the settings). The results of CNN+AGN and Bert+AGN are depicted in Figure 2c, where the curves show that combining TCoL from Yelp P. can produce substantial improvements to both original models, i.e., around 2% on SST-2 dataset. Meanwhile, we notice that the best results are achieved with relatively larger values of ϵ when exploiting TCoL from Yelp P. ($\epsilon = 0.40$ for Bert+AGN and $\epsilon = 0.35$ for CNN+AGN) than that for models using TCoL of SST-2 only ($\epsilon = 0.20$ for Bert+AGN and $\epsilon = 0.10$ for CNN+AGN, similar patterns can also be found in Figure 2a and Figure 2b). A larger ϵ means the classifier can depend more on TCoL because of the improved trustworthiness of statistical feature, and vice versa. We can conclude that, although the improvements can be compromised due to the bias caused by data size, we can

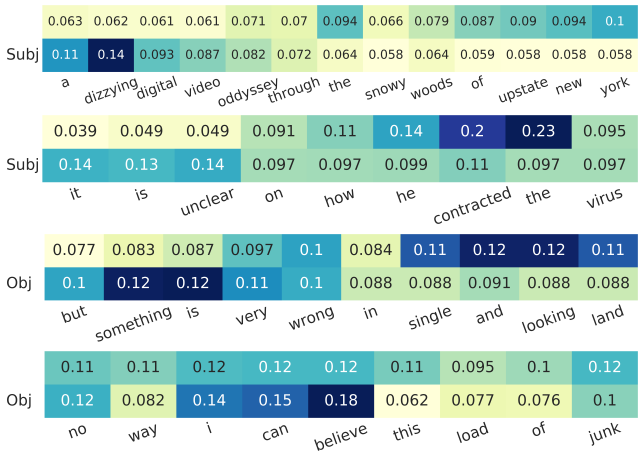


Figure 3: Heatmaps of attention weights for models w/o (upper) and w/ (lower) statistical information. "Subj" and "Obj" represent the ground-truth label of each sentence.

still apply the TCoL from a large dataset to a smaller one as compensation to boost the model performance.

5.3 Adversarial attacks on TCoL

The previous section has discussed the bias of TCoL caused by data amount due to the deviated prior distribution, where the prior distribution is generally correct and can be compensated by introducing more massive datasets. However, under some extreme circumstances, prior knowledge from TCoL can be partially correct or totally incorrect due to manual manipulation. To verify the robustness of AGN against deliberately corrupted statistical information, we conduct adversarial attacks on two datasets by shuffling the TCoL dictionary. For each trial, we randomly shuffle 10%, 50%, and 100% of the whole TCoL dictionary. The results are shown in Table 4. In general, AGN with adaptive valve ($\epsilon = 0.2$) can avoid severe performance deterioration compared with model accepting all statistical information ($\epsilon = 0.5$) when the TCoL is thoroughly shuffled. Moreover, with 50% and 10% shuffled statistical information, CNN+AGN ($\epsilon = 0.5$) achieves lower results compared with CNN+S/A on SST-2, while CNN+AGN ($\epsilon = 0.2$) can outperform CNN+S/A. This observation shows that the manipulation of statistical information does cause noise to the classifier and compromise the performance. Meanwhile, it also proves that the proposed AGN is robust enough against such attacks.

5.4 VAE *vs* AE in the V-Net

Both VAE and AE are both powerful representation learning models. In this section, we compare the performance of using TCoL encoded by variational autoencoder in the V-Net and that encoded by vanilla autoencoder. We conduct additional experiments on two small datasets and two large datasets; the results are reported in Table 5. We notice that, although the improvements brought by using variational inference are sometimes marginal, the VAE-based AGN consistently outperforms AE-based AGN in both metrics with a more stable performance. Especially, we observe improvements of 0.49% in F1 score on Yelp F. and 0.62% in accuracy on Yelp P. These results show that bounding latent space with a prior distribution is beneficial to the representation learning for statistical information, which is conducive to enhance the classification performance.

Shuffle Rate		100%	50%	10%
SST-2	$\epsilon = 0.2$	80.05	81.13	82.00
	$\epsilon = 0.5$	78.25	79.52	79.79
TREC	$\epsilon = 0.2$	89.64	90.43	90.88
	$\epsilon = 0.5$	87.69	88.89	89.00

Table 4: Adversarial attack on SST-2 and TREC datasets. The results are accuracy (%) of experiments on CNN+AGN.

	VAE		AE	
	Accu.	F1	Accu.	F1
SST-2	82.24	82.26	81.98	82.20
TREC	92.11	90.75	92.02	90.64
Yelp F.	95.15	93.76	94.89	93.36
Yelp P.	67.93	59.39	67.11	59.01

Table 5: Comparison between using VAE and AE as TCoL encoder on SST-2, TREC, and Yelp datasets. The results (%) are of experiments on CNN+AGN.

5.5 Scalability of Implementation

The generation of TCoL is scalable since we only need to traverse through the corpus once, which is of linear time complexity $\mathcal{O}(n)$, where n denotes the total number of words in the corpus. For the model training process, the V-Net is a standard VAE/AE, and the number of parameters in S-Net is not significantly increased compared with the corresponding baseline model. For example, a CNN+AGN only requires $3,250$ additional parameters and 0.13 second more per epoch on training time, compared with a standard TextCNN (on SST-2 with an RTX 2080 Ti GPU). Therefore, the proposed AGN will neither require much higher computational power nor much more computation time than the widely adopted baseline models under practical settings.

6 Conclusion & Future work

In this paper, we have proposed an Adaptive Gate Network (AGN) to incorporate statistical features and conducted extensive experiments with CNN-based, RNN-based, Transformer-based, and Bert-based frameworks to demonstrate the effectiveness and robustness of our proposed AGN method. The well-designed valve mechanism enables AGN to merge necessary information while preserving essential semantic features. Given naturally biased statistical information, AGN can produce impressive improvements to baseline models. AGN has excellent flexibility in actual usage, especially when limited external information is available. For future work, we plan to apply the AGN to other tasks exploring various types of external knowledge, thereby providing a more in-depth insight into the framework design with better utilization of additional knowledge.

7 Acknowledgments

Xianming Li's work has been supported by Ant Group. Zongxi Li's work has been supported by City University of Hong Kong. Haoran Xie's work has been supported by the Faculty Research Fund (102041) and the Lam Woo Research Fund (LWI20011) at Lingnan University, Hong Kong. Qing Li's work has been supported by a general research fund from the Hong Kong Research Grants Council (project number: PolyU 112114/17E). We thank the anonymous reviewers for their careful reading of our manuscript and their insightful comments and suggestions.

References

Aizawa, A. 2003. An Information-Theoretic Perspective of TF–iDF Measures. *Information Processing & Management* 39(1): 45–65.

Bahdanau, D.; Cho, K.; and Bengio, Y. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *Proceedings of the 3rd International Conference on Learning Representations*.

Chen, J.; Hu, Y.; Liu, J.; Xiao, Y.; and Jiang, H. 2019. Deep Short Text Classification with Knowledge Powered Attention. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*.

Chung, J.; Gülçehre, Ç.; Cho, K.; and Bengio, Y. 2014. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *CoRR* abs/1412.3555.

Conneau, A.; Schwenk, H.; Barrault, L.; and LeCun, Y. 2017. Very Deep Convolutional Networks for Text Classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*.

Devlin, J.; Chang, M.; Lee, K.; and Toutanova, K. 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*, 4171–4186.

Graves, A.; Jaitly, N.; and Mohamed, A.-r. 2013. Hybrid Speech Recognition with Deep Bidirectional LSTM. In *Proceedings of 2013 IEEE Workshop on Automatic Speech Recognition and Understanding*.

Hochreiter, S.; and Schmidhuber, J. 1997. Long Short-Term Memory. *Neural Computing* 9(8): 1735–1780. ISSN 0899-7667. doi:10.1162/neco.1997.9.8.1735.

Johnson, R.; and Zhang, T. 2017. Deep Pyramid Convolutional Neural Networks for Text Categorization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*.

Kim, Y. 2014. Convolutional Neural Networks for Sentence Classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*.

Kingma, D. P.; and Welling, M. 2014. Auto-Encoding Variational Bayes. In *Proceedings of the 2014 International Conference on Learning Representations*.

Lai, S.; Xu, L.; Liu, K.; and Zhao, J. 2015. Recurrent Convolutional Neural Networks for Text Classification. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*.

Lan, Z.; Chen, M.; Goodman, S.; Gimpel, K.; Sharma, P.; and Soricut, R. 2020. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. In *Proceedings of the 8th International Conference on Learning Representations*.

Li, X.; and Roth, D. 2002. Learning Question Classifiers. In *Proceedings of the 19th international conference on Computational linguistics*.

Liang, W.; Xie, H.; Rao, Y.; Lau, R. Y.; and Wang, F. L. 2018. Universal Affective Model for Readers' Emotion Classification Over Short Texts. *Expert Systems with Applications* 114: 322 – 333.

Pang, B.; and Lee, L. 2004. A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*.

Post, M.; and Bergsma, S. 2013. Explicit and Implicit Syntactic Features for Text Classification. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*.

Ramos, J.; et al. 2003. Using TF-iDF to Determine Word Relevance in Document Queries. In *Proceedings of the first instructional conference on machine learning*.

Rojas, K. R.; Bustamante, G.; Oncevay, A.; and Sobrevilla Cabezudo, M. A. 2020. Efficient Strategies for Hierarchical Text Classification: External Knowledge and Auxiliary Tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Salton, G.; and Buckley, C. 1988. Term-Weighting Approaches in Automatic Text Retrieval. *Information processing & management* 24(5): 513–523.

Socher, R.; Perelygin, A.; Wu, J.; Chuang, J.; Manning, C. D.; Ng, A.; and Potts, C. 2013. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*.

Tang, D.; Qin, B.; and Liu, T. 2015. Document Modeling with Gated Recurrent Neural Network for Sentiment Classification. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*.

Tang, H.; Ji, D.; Li, C.; and Zhou, Q. 2020. Dependency Graph Enhanced Dual-transformer Structure for Aspect-based Sentiment Classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Teng, Z.; Vo, D.-T.; and Zhang, Y. 2016. Context-Sensitive Lexicon Features for Neural Sentiment Analysis. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is All you Need. In *Proceedings of Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, 5998–6008.

Wang, J.; Wang, Z.; Zhang, D.; and Yan, J. 2017. Combining Knowledge with Deep Convolutional Neural Networks for Short Text Classification. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*.

Yang, Y.; and Pedersen, J. O. 1997. A Comparative Study on Feature Selection in Text Categorization. In *Proceedings of the International Conference on Machine Learning*.

Yang, Z.; Yang, D.; Dyer, C.; He, X.; Smola, A.; and Hovy, E. 2016. Hierarchical Attention Networks for Document Classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Zhang, X.; Zhao, J. J.; and LeCun, Y. 2015. Character-level Convolutional Networks for Text Classification. In *Proceedings of Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015*, 649–657.