Clickbait Detection via Contrastive Variational Modelling of Text and Label

Xiaoyuan Yi\textsuperscript{1}, Jiarui Zhang\textsuperscript{2}, Wenhai Li\textsuperscript{2}, Xiting Wang\textsuperscript{1} and Xing Xie\textsuperscript{1}
\textsuperscript{1}Microsoft Research Asia
\textsuperscript{2}Tsinghua University
\{xiaoyuanyi, xitwan, xing.xie\}@microsoft.com, \{zhangjr18, wh-li20\}@mails.tsinghua.edu.cn

Abstract

Clickbait refers to deliberately created sensational or deceptive text for tricking readers into clicking, which severely hurts the web ecosystem. With a growing number of clickbaits on social media, developing automatic detection methods becomes essential. Nonetheless, the performance of existing neural classifiers is limited due to the underutilization of small labelled datasets. Inspired by related pedagogy theories that learning to write can promote comprehension ability, we propose a novel Contrastive Variational Modelling (CVM) framework to exploit the labelled data better. CVM models the conditional distributions of text and clickbait labels by predicting labels from text and generating text from labels simultaneously with Variational AutoEncoder and further differentiates the learned spaces under each label by a mixed contrastive learning loss. In this way, CVM can capture more underlying textual properties and hence utilize label information to its full potential, boosting detection performance. We theoretically demonstrate CVM as learning a joint distribution of text, clickbait label, and latent variable. Experiments on three clickbait detection datasets show our method’s robustness to inadequate and biased labels, outperforming several recent strong baselines.

1 Introduction

Driven by economic benefits, online content publishers may deliberately craft alluring or deceptive content in headlines, posts, and hashtags, known as clickbait, to entice readers to click the accompanying links, as shown in Figure 1. Readers are lured by the curiosity gap [Indurthi et al., 2020] into clicking and then usually led to uninformative or unrelated content. These clickbaits could bring serious problems such as hurting users’ reading experience, obstructing retrieval / recommendation algorithms, shaping and spreading wrong public opinions, eroding user trust and publishers’ reputation, and eventually damaging the whole web ecosystem [Ecker et al., 2014]. With the prosperity of social media, there has been consistent growth in the number of clickbaits [Rony et al., 2017], posing an urgent need for developing effective automatic clickbait detection methods.

The research paradigm of clickbait detection evolved from early feature engineering-based methods [Biyani et al., 2016; Chakraborty et al., 2016; Wei and Wan, 2017] to neural networks [Yoon et al., 2019; Mishra et al., 2020] and, more recently, into pre-trained language models like BERT [Devlin et al., 2019] which also show superiority in this task [Indurthi et al., 2020]. However, these models suffer from inadequate labels since they act as simple discriminative classifiers and ignore other helpful information [Bishop and Lasserre, 2007], causing a spurious overfitted boundary [Yogatama et al., 2017].

To address this problem, we investigate how to better exploit labelled datasets. Relevant pedagogy theories manifest that writing learning could help create meaningful text representations and enhance comprehension ability [Caccamise, 2011]. Inspired by this, we consider promoting the model’s understanding of what clickbait looks like by learning to write it. Concretely, we model the joint distribution of text (e.g., news headlines) and labels (clickbait or not), namely a generative model, which has proven to be capable of capturing underlying data properties beyond labels, more robust to label bias and superior to the discriminative ones with limited data [Ng and Jordan, 2002; Yogatama et al., 2017].

For this sake, we propose a novel Contrastive Variational Modelling (CVM) framework. CVM learns the joint distri-
tion by characterizing two symmetric optimization directions, predicting labels from text and generating text from given labels, as a dual process and optimizing them simultaneously with each direction modelled via a Variational AutoEncoder (VAE) [Kingma and Welling, 2014]. The latent space in VAE equips CVM with flexible representations [Ding and Gimbel, 2019], the generation direction captures richer textual properties, and the prediction one better differentiates the text space with label information. Besides, we design a mixed contrastive learning loss to further disentangle both the text and latent spaces, which helps align representations with corresponding labels. Our method can be theoretically regarded as learning a joint distribution of text, clickbait label, and latent variable and simultaneously minimizing the uncertainty of labels given text and, inversely, the uncertainty of text given labels. By this means, CVM can build closer connections between text and labels, learn more distinguishable representations and hence utilize the limited labelled data to their full potential, further improving detection performance.

In summary, the contributions of this paper are as follows:

- We propose a novel contrastive variational modelling method to build more meaningful text representations via learning to write, benefiting clickbait detection.
- We interpret CVM as approximating the real joint distribution of text, clickbait label, and latent variable, and minimizing the uncertainty of prediction and generation, providing a theoretical guarantee for our method.
- We conduct experiments on three datasets and demonstrate that CVM can outperform several recent strong baselines like BERT when used for fine-tuning and achieve comparable results when trained from scratch.

2 Related Work

2.1 Clickbait Detection

The rampant clickbait on social media have raised increasing research interest in automatic clickbait detection. Early attempts focus on feature engineering and consider various features such as linguistics [Chakraborty et al., 2016], style [Biyani et al., 2016; Wei and Wan, 2017], semantics [Rony et al., 2017] and multi-modal features [Ha et al., 2018]. Then neural networks, e.g., Recurrent Neural Networks (RNN), which have flourished over the past decade, are also utilized for this challenging task [Anand et al., 2017]. Mishra et al. [2020] further use cross-attention to calculate the similarity of news headlines and corresponding body text, and then tackle clickbaitings containing cardinal values by part-of-speech tag patterns [Mishra and Zhang, 2021].

Despite notable advantages, these neural models still suffer from limited labelled data. One possible solution is data augmentation [Yoon et al., 2019] which produces adequate pseudo clickbait text with specified labels, while the unstable quality of pseudo data may also bring additional noise.

More recently, pre-trained language models (PLMs) [Devlin et al., 2019; Dong et al., 2019] have made a breakthrough for both natural language understanding (NLU) and generation (NLG) due to implicit knowledge learned from massive plain text, which also helps clickbait detection [Indurthi et al., 2020]. However, we believe these models’ performance can be further improved since as simple discriminative classifiers, they don’t fully exploit labelled data, as discussed in Sec. 1.

2.2 Combining NLU with NLG

The idea of combining understanding with generation can be traced back to the age of statistical machine learning [McCallum et al., 2006], which still keeps developing. The recent proposed Dual Learning (DL) [Xia et al., 2017] pairs two tasks, where the input of one task is the target of the other, to exploit the duality and benefit both. For example, image classification and image generation, sentiment analysis and sentimental text generation. DL trains two separate models for each task and builds the connection by a regularization term derived from Lagrange multipliers. Moreover, DL requires multiple models and optimizing marginal distributions, which is too resource-consuming, especially for large PLMs.

2.3 VAE for NLU

Due to the ability of learning a more flexible latent space, VAE has shown the superiority of both image [Kingma and Welling, 2014] and text [Fu et al., 2019] generation, leading to several VAE-based works relevant to our model. Tseng et al. [2020] apply DL to generating restaurant descriptions given attributes like name, food, and rating, and then couple it with attribute words extraction from descriptions (as a kind of understanding). The two tasks share a latent space to boost each other. OPTIMUS [Li et al., 2020] takes a pre-trained BERT as the encoder and GPT-2 [Radford et al., 2019] as the decoder and then continues pre-training them as a VAE. VAMPIRE [Gururangan et al., 2019] simplifies the encoder and decoder of VAE as Multi-Layer Perceptron (MLP) with bag-of-words inputs and then uses the hidden states from multiple layers for text classification. Cheng et al. [2019] jointly train a VAE for text generation and a classifier for classification, which also provides pseudo labels for unlabelled data during training. This model can be regarded as a kind of multi-task learning which combines two relevant tasks.

Our motivation and method considerably differ from aforementioned work. To fully exploit labelled clickbait data, we recourse generation learning to provide more expressive textual representations, and model the prediction as fitting a symmetrical label distribution via VAE (instead of simply training an individual classifier and adding up the losses). The two directions share the same Transformer parameters but incorporate different prior latent spaces, which are further distinguished by a mixed contrastive loss. We also provide a theoretical interpretation of the whole training objective.

3 Methodology

3.1 Formalization and Overview

Before detailing the proposed CVM, we first formalize our task. Define $x$ as the text to detect, e.g., news headlines or Tweet posts, $c$ as the linked contents like news body, and $y$ as the label of clickbait ($y = 1$) or not ($y = 0$). Our goal is to correctly predict $y$ in terms of only textual $x$ and $c$, which is conventionally characterized as a discriminative classifier $q(y|x, c)$ [Indurthi et al., 2020]. Unlike most methods, we
propose to learn a joint distribution \( p(y, x, c) \) to capture more meaningful textual properties belonging to each label and therefore build a better connection between text and labels.

For this purpose, we simultaneously take two optimization directions: generating text from given clickbait labels \( q(x|y, c) \) and predicting labels according to text \( q(y|x, c) \). The former helps the model understand what clickbait and non-clickbait contents look like, and the latter learns distinguishable spaces of each label and helps differentiate the text space. Both directions are modelled via VAE to learn more flexible representations. We also design a mixed contrastive loss to further separate text \( (x) \) and latent variables \( (z) \) under different labels. Figure 2 depicts the architecture of CVM.

3.2 Basic Structure

We use a Transformer [Vaswani et al., 2017] decoder as the backbone of CVM, which consists of \( L \) stacked layers and transfers a text sequence to contextualized hidden states. In \( l \)-th layer, states \( h^l \) are calculated by multi-head self-attention:

\[
\text{Attention}(Q, K, V) = \text{softmax} \left( \frac{QK^T}{\sqrt{d}} \right) V,
\]

where \( Q, K, V \) represent query, key and value, respectively, which are projected from hidden states of the previous layer: \( Q; K; V = H^{l-1}W_q^l; H^{l-1}W_k^l; H^{l-1}W_v^l \).

When this decoder serves as the feature extractor, we place a [CLS] token at the beginning of the given \( x \), and use its hidden state in the last layer as the corresponding representation \( h_x \). When using it as a generator, we apply a causal mask to the self-attention to conduct autoregressive generation. All components of CVM share the same Transformer decoder.

3.3 Variational Modelling

For modelling text generation \( q(x|y, c) \), we use a conditional VAE and maximize its evidence lower bound (ELBO) as:

\[
\log q(x|y, c) \geq \mathbb{E}_{p(z|x,c,y)}[\log q_g(x|z, c, y)] - \text{KL}[p(z|x, c, y)||q_g(z|c, y)] = -L_g,
\]

where \( z \) is the latent variable, the true prior and posterior distributions of \( z \) are approximated with a prior network \( q_G(z|c, y) \) and a recognition network \( p(z|x, c, y) \) respectively, and \( q_g(x|z, c, y) \) is the decoder to generate text. Similarly, we maximize the ELBO of the label distribution:

\[
\log q(y|x, c) \geq \mathbb{E}_{p(z|x,c,y)}[\log q_c(z|y, c, x)] - \text{KL}[p(z|x, c, y)||q_c(z|y, c, x)] = -L_c.
\]

Detailed derivations are presented in Appendix A.

Different from combining classification and generation losses in a simple multi-task learning manner [Cheng et al., 2019], the two optimization directions above are symmetric, which allows CVM to approximate a desired joint distribution directly (Theorem 1), separate the text space better (Figure 4), and achieve further improvement (Table 2).

In detail, we assume the latent variable follows the isotropic Gaussian distribution in accordance with previous work [Kingma and Welling, 2014; Fu et al., 2019], and then implement the recognition and prior networks with MLP, e.g., for label prediction: \( \mu_{pri}; \log(\sigma_{pri}) = \text{MLP}(h_x, h_c), \quad [\mu_{pos}; \log(\sigma_{pos})] = \text{MLP}(h_x, h_c, e_y) \), where \( e_y \) and the subscripts \( pri \) and \( pos \) means label embedding, prior and posterior respectively. Then we can get samples of latent variables with the reparametrization trick [Kingma and Welling, 2014]: \( z = \mu + \sigma \epsilon, \epsilon \sim \mathcal{N}(0, 1) \), where \( z \) is sampled from the posterior distribution for training and the prior one for testing.

We predict the probability of each label by \( q_c(y|z, x, c) \), and achieve further improvement (Table 2). In detail, we assume the latent variable follows the isotropic Gaussian distribution in accordance with previous work [Kingma and Welling, 2014; Fu et al., 2019], and then implement the recognition and prior networks with MLP, e.g., for label prediction: \( \mu_{pri}; \log(\sigma_{pri}) = \text{MLP}(h_x, h_c), \quad [\mu_{pos}; \log(\sigma_{pos})] = \text{MLP}(h_x, h_c, e_y) \), where \( e_y \) and the subscripts \( pri \) and \( pos \) means label embedding, prior and posterior respectively. Then we can get samples of latent variables with the reparametrization trick [Kingma and Welling, 2014]: \( z = \mu + \sigma \epsilon, \epsilon \sim \mathcal{N}(0, 1) \), where \( z \) is sampled from the posterior distribution for training and the prior one for testing.

We predict the probability of each label by \( q_c(y|z, x, c) \), and achieve further improvement (Table 2). In detail, we assume the latent variable follows the isotropic Gaussian distribution in accordance with previous work [Kingma and Welling, 2014; Fu et al., 2019], and then implement the recognition and prior networks with MLP, e.g., for label prediction: \( \mu_{pri}; \log(\sigma_{pri}) = \text{MLP}(h_x, h_c), \quad [\mu_{pos}; \log(\sigma_{pos})] = \text{MLP}(h_x, h_c, e_y) \), where \( e_y \) and the subscripts \( pri \) and \( pos \) means label embedding, prior and posterior respectively.

We predict the probability of each label by \( q_c(y|z, x, c) \), and achieve further improvement (Table 2). In detail, we assume the latent variable follows the isotropic Gaussian distribution in accordance with previous work [Kingma and Welling, 2014; Fu et al., 2019], and then implement the recognition and prior networks with MLP, e.g., for label prediction: \( \mu_{pri}; \log(\sigma_{pri}) = \text{MLP}(h_x, h_c), \quad [\mu_{pos}; \log(\sigma_{pos})] = \text{MLP}(h_x, h_c, e_y) \), where \( e_y \) and the subscripts \( pri \) and \( pos \) means label embedding, prior and posterior respectively.

3.4 Mixed Contrastive Loss

The unbalanced label distribution may impede the separation of text space. To build a tighter connection between text and clickbait labels, and disentangle the space into label-conditioned subspaces, we maximize the conditional mutual information \( I(x, y|c) \). Benefiting from CVM’s generation ability (Eq.(2)), we can generate counterparts \( x \) from each \( c \) with the opposite clickbait label, which allows us to approximate the intractable mutual information by InfoNCE.
\[ \mathcal{L}_{info} = -\log \frac{\sum_{v_1, v_2 \in S_+} \exp(d(v_1, v_2)/\tau)}{\sum_{v_1, v_2 \in S_+ \cup S_-} \exp(d(v_1, v_2)/\tau)}, \]

where \( S_+ \) is the set of positive pairs \((v_1, v_2)\) which can be \((x, y)\) or \((x^+, y)\) with \(x^+\) sampled from \(q_g(x|z, c, y)\); \( S_- \) is the set of negative pairs \((x^-, y)\) with those sampled from \(q_g(x|z, c, 1-y)\); \( \tau \) is a hyperparameter. \( d \) is the distance calculated as \(d(x, y) = (h, W^t)^\top e_y / ||h, W^t|| ||e_y||\), where \( W^t \) is a learnable parameter matrix to project one representation \( e_y \), into the same dimension as the other, e.g., \( e_y \).

However, sampling generated text during training is quite time-consuming. To accelerate training, we design a mixed contrastive loss to involve various combinations of pairs. Concretely, we generate only one \( z^+ \) and one \( z^- \) for each tuple \((c, y)\), and also add \((z^+, y), (x, x^+)\) into \( S_+ \), and \((z^-, 1-y), (x, x^-)\) into \( S_- \), where \( z^+ \sim q_g(z|y, c) \) and \( z^- \sim q_g(z|1-y, c) \). We use \( K \) positive latent samples \( z^+ \) and \( K \) negative ones \( z^- \). In this way, we separate the learned text space and the latent space under different labels from which the text one is derived, as shown in Figure 2.

3.5 Training Objective and Prediction Method

Based on the components introduced above, we minimize the total training objective:

\[ \mathcal{L}_{CVM} = \mathbb{E}_{p(x,c,y)}[\mathcal{L}_c + \mathcal{L}_g + \mathcal{L}_{info} + \mathcal{L}_{cond}], \]

where \( \mathcal{L}_{cond} = -\log q_g(x|c) \) is a simple conditional generation loss, e.g., generating news headlines from news bodies; \( \hat{p}(x, c, y) \) is the empirical distribution (training set).

Then we give the following conclusion:

**Theorem 1.** The training objective \( \mathcal{L}_{CVM} \) is an approximate upper bound of \( KL[p(x,c,y,z)||q(x,c,y,z)] + H(y|x,c) + H(x|y,c) \), where \( p \) is the real joint distribution, \( q \) is the estimated one, and \( H \) is the conditional Shannon entropy.

**Proof.** See Appendix B.

Theorem 1 demonstrates that CVM actually learns an estimated joint distribution \( q(x,c,y,z) \) which approximates the real one, and meanwhile reduces the uncertainty of both classification and generation, coinciding with our initial motivation and providing a guarantee of our model’s effectiveness.

For clickbait label detection, in the training phase, we directly use \( q_c(y|x, z, c) \) with \( z \) sampled from \( p(z|x, c, y) \) to enhance the robustness of CVM through the stochastic \( z \). In the testing phase, we can sidestep the randomness by \( q_c(y|x, z, c) = \int q_c(y|x, z, c) q(z|x, c) dz \approx \frac{1}{M} \sum_{m=1}^{M} q_c(y|x, z, c) z^m \sim q_c(x|z, c) \), while such Monte Carlo method is slow and unstable. Therefore we use the prior mean vector \( \mu_{p(y|x)} \) instead of \( z \). We will show the more efficient latter one performs comparably in Sec. 4.5.

4 Experiments

4.1 Data & Metrics

We conduct experiments on three clickbait-related datasets.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Training</th>
<th>Validation</th>
<th>Testing</th>
</tr>
</thead>
<tbody>
<tr>
<td>News</td>
<td>17,538</td>
<td>1,500</td>
<td>3,063</td>
</tr>
<tr>
<td>Tweet</td>
<td>17,588</td>
<td>2,000</td>
<td>17,554</td>
</tr>
<tr>
<td>NELA</td>
<td>50,000</td>
<td>6,690</td>
<td>6,745</td>
</tr>
</tbody>
</table>

Table 1: Data Statistics. Ratios in brackets indicate the proportion of positive labels (clickbait or incongruent).

News Clickbait Detection (News). A public Kaggle competition dataset for news headline clickbait detection\(^2\). Since the original testing labels are unavailable, we merge the others and re-split them into training, validation and testing sets.

Tweet Clickbait Detection (Tweet). A multi-modal dataset for the Tweet posts clickbait detection competition\(^1\), which contains the image, post time, post text, title and paragraphs of the linked article, etc. Each post is annotated with five clickbait strength scores by five annotators. We set the label to 1 when the average score > 0.5 otherwise 0, and use only post text, title and paragraphs for our experiments.

News Headline Incongruence Detection (NELA). An automatically constructed dataset for detecting incongruity between a given news headline and body text [Yoon et al., 2019], where each headline is combined with an irrelevant body. We use the original validation and testing sets, and 50,000 randomly selected training samples as the training set.

Table 1 presents detailed data statistics.

**Metrics.** We take clickbait detection as binary classification and report accuracy, Macro F1 and ROC AUC score.

4.2 Setups

We use a BERT-like Transformer encoder with 12 layers, hidden size 768 and 8 attention heads as the basic structure of our model, and initialize it with a pre-trained Unilm [Dong et al., 2019]. The label embedding size, latent variable size, number of latent samples \( K \), batch size and learning rate are 64, 256, 16, 24 and 2e-4, respectively. We use cyclic annealing [Fu et al., 2019] to alleviate the KL annealing problem in VAE training. We report the average results over five runs. More setting details are provided in Appendix C.

4.3 Baselines

We conduct comprehensible comparisons with the following six models. **BILSTM** [Anand et al., 2017]: a bidirectional LSTM classifier for clickbait detection. **MuSeM** [Mishra et al., 2020]: a very recent cross attention based model for incongruity detection, which calculates semantic matching between the original headline and the synthetic one. **AS-VAET** [Cheng et al., 2019]: a Transformer-based classifier that trains the Transformer as a VAE to provide text representations for the jointly trained classifier, which can be considered as a simple kind of multi-task training. **VAMPIRE** [Gururangan et al., 2019]: this model first pre-trains a VAE, in which the encoder and decoder are simplified as MLP, and then uses the hidden states from multiple layers for text classification. For fair comparisons, we replace their simple MLP

---

\(^1\)Oord et al. prove that the negative InfoNCE loss is a lower bound of mutual information.

\(^2\)https://www.kaggle.com/c/clickbait-news-detection

\(^3\)https://webis.de/events/clickbait-challenge
with a pre-trained 12-layer GPT-2. BERT [Devlin et al., 2019]: a fine-tuned 12-layer BERT, which proves to outperform most previous models on clickbait detection [Indurthi et al., 2020]. UniLM [Dong et al., 2019]: we also fine-tuned a 12-layer UniLM that is pre-trained for both NLU and NLG.

### 4.4 Experimental Results

Table 2 shows that our model achieves the best performance on most metrics over the three datasets. The fine-tuned BERT and UniLM get much better results than other baselines, benefiting from the knowledge learned from a vast number of plain text. However, acting as simple discriminative classifiers, these big models would overfit after several epochs of fine-tuning, limiting their performance. On the contrary, CVM further improves F1 and AUC compared to BERT and UniLM, supporting our claim that CVM can fully exploit the labelled data by learning more meaningful textual properties. All models perform worse on Tweet since this multi-modal dataset is more challenging, and some clickbait could be correctly identified only when other information, e.g., image and metadata, is taken into account, while we use merely post and body text. Even so, CVM still notably improves F1.

Another interesting result is that multi-task-style ASV AET is inferior to VAMPIRE on both News and Tweet, which indicates that improper combination with generation could hamper classification. Moreover, its unreliable (lower F1) classifier may produce noisy labels for unlabelled data, bringing negative signals. Additionally, on NELA, models with the attention mechanism like ASV AET significantly outperform the others like VAMPIRE. This is because the incongruity between a headline and a randomly combined body can be captured by attention more easily, while these authentic headlines contain no clickbait patterns, which hinders those relying on extracted representations but not semantic matching.

Besides, we find our model gets less improvement (0.9 F1) on NELA compared to News (1.9 F1) and Tweet (1.2 F1). The reason lies in that NELA contains more (almost three times) and balanced training instances than the other two, and as a generative model, CVM is more effective on small and biased datasets, as mentioned in Sec. 1, manifesting the suitability of CVM for data-limited clickbait detection.

### 4.5 Further Analysis

#### Ablation Study
As shown in Table 3, without the generation loss \( L_g \) in Eq.(5), the performance of CVM could drop a lot (2.9 F1), much worse than BERT and UniLM, verifying our claim that CVM can benefit clickbait detection by learning to write. In spite of the performance loss, CVM is still superior to other VAE-based models like ASV AET and VAMPIRE. Such a result suggests that rather than simply combining an individual classifier or using the pre-trained VAE as a feature extractor, a better way to incorporate VAE for classification is approximating the label distribution as Eq.(3).

Besides, we can see the conditional generation loss \( L_{\text{cond}} \) makes an insignificant difference (0.5 F1 drop). By contrast, the contrastive samples, both text and latent variables in Eq.(4), bring a remarkable improvement (2.4 F1), and the latent ones play a more critical role. This is because our mixed contrastive loss \( L_{\text{info}} \) can help CVM lean a more expressive and distinguishable text space (See Figure 4 (c)).

More interestingly, we also try to train CVM from scratch without UniLM initialization and find it achieves comparable F1 and AUC scores (but lower accuracy) to the two fine-tuned PLMs, which indicates that our model is insensitive to data size to some extent and more suitable for the scenario of

<table>
<thead>
<tr>
<th>Method</th>
<th>News</th>
<th>Tweet</th>
<th>NELA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Accuracy</td>
<td>Macro F1</td>
<td>AUC</td>
</tr>
<tr>
<td>CVM (Ours)</td>
<td>81.3</td>
<td>71.4</td>
<td>88.5</td>
</tr>
</tbody>
</table>

Table 3: Ablation study on News.
limited data like clickbait detection, as delineated in Sec. 1.

Precision-Recall Curve. In Figure 3, we draw the precision-recall curve of different PLM-based models on News. We can find that CVM gets higher recall as well as satisfactory precision. Higher recall means our model could cover and identify more potential clickbaits for further re-checking by human inspectors, which is more suitable for clickbait detection. In addition, PLMs give extremely higher probabilities for their predictions, e.g., \(p(y|x, c) > 0.95\), while our CVM can produce smoother probabilities, which allow higher precision, yielding more flexible choices of probability thresholds for various real application scenarios.

Effect of Number of Latent Samples. In Figure 4 (a), we investigate the influence of different numbers (\(K\)) of sampled latent variables in Eq.(4). With the increase of \(K\), F1 will increase first and then decrease, reaching the maximum at \(K = 16\). We think this is mainly because too few samples (e.g., \(K = 4\)) are insufficient for the mixed contrastive loss to distinguish latent spaces under different labels, while too many samples could also involve outliers far from the centre, causing much noise and hurting performance.

In Figure 4 (b), we compare the two methods to calculate prediction probabilities \(p_c(y|x, c)\) in Sec. 3.5, using Monte Carlo sampling or mean vector \(\mu_{pri}\). Using Monte Carlo, F1 increases first and then converges with more than 128 samples, which is memory-consuming. In contrast, using \(\mu_{pri}\) for prediction leads to similar performance and is more efficient.

Visualization of Text and Latent spaces. In Figure 4 (c), we visualize text representations (\(h_c\)) of different models and prior latent variables sampled from CVM. We can observe that the original pre-trained BERT fails to distinguish headlines with different labels, while the fine-tuned BERT and UniLM split them into two sub-spaces, though there are still many mixed ones. Compared to these PLMs, CVM learns a more complex and distinguishable space which is further differentiated by the contrastive loss, benefiting clickbait detection. We also plot samples from the latent space of CVM, which provides smoother representations for prediction.

News Body: Super Bowl LI is only a few days away. Ads 30-second ads costs between $5 million and $5.5 million this year... Super Bowl commercials used to be bargain to reach 100 million viewers... It’s halftime performer Lady Gaga who gets hurt... Members of the winning team get $107,000 each...

Generated Headline. Table 4 gives some examples generated by our model, showing that CVM can produce plausible news headlines that help learn distinguishable text representations. We did not evaluate generation since our focus is not NLG. More generated cases are listed in Appendix D.

4.6 Conclusion and Future Work

We propose CVM, a contrastive variational modelling framework to boost clickbait detection performance via learning to write. CVM fuses label prediction and text generation by VAE, and learns distinguishable representations via a mixed contrastive loss. In this way, our model can be theoretically regarded as approximating the real joint distribution of text, clickbait label, and latent variable. Experiments show that CVM is more robust to small and biased datasets with higher recall, more suitable for real clickbait detection scenarios.

In the future, we plan to explore the potential to benefit controllable generation via learning to classify and extend our method to multiple classification and generation tasks.
References


