Text Classification through Glyph-aware Disentangled Character Embedding and Semantic Sub-character Augmentation

Takumi Aoki **Shunsuke Kitada** **Hitoshi Iyatomi**
Department of Applied Informatics, Graduate School of Science and Engineering
Hosei University, Tokyo, Japan
{takumi.aoki.4g, shunsuke.kitada.8y}@stu.hosei.ac.jp
iyatomi@hosei.ac.jp

Abstract

We propose a new character-based text classification framework for non-alphabetic languages, such as Chinese and Japanese. Our framework consists of a variational character encoder (VCE) and character-level text classifier. The VCE is composed of a β-variational auto-encoder (β-VAE) that learns the proposed glyph-aware disentangled character embedding (GDCE). Since our GDCE provides zero-mean unit-variance character embeddings that are dimensionally independent, it is applicable for our interpretable data augmentation, namely, semantic sub-character augmentation (SSA). In this paper, we evaluated our framework using Japanese text classification tasks at the document- and sentence-level. We confirmed that our GDCE and SSA not only provided embedding interpretability but also improved the classification performance. Our proposal achieved a competitive result to the state-of-the-art model while also providing model interpretability.

1 Introduction

Some Asian languages (e.g., Chinese and Japanese) use *glyphs* to give visual meaning to characters. For example, the following Japanese characters have a common form of "⻌," which is a sub-character meaning of the related word *road*: "迫" (approach: come near the destination by *road*) and "追" (follow: track the *road*). In consideration of these characteristics of the language, several glyph-aware natural language processing (NLP) models have been proposed (Shimada et al., 2016; Liu et al., 2017; Kitada et al., 2018; Sun et al., 2019). These deep-learning-based models train input text as a sequence of character images and learn character embeddings from the images.

In general, the interpretability of the NLP model is important in terms of its reliability, as well as providing the required performance for the task. If imaged-based models can learn these sub-characters in a way that is interpretable, it helps greatly in improving the overall interpretability of the models.

In terms of improving the interpretability of models, disentangled representation learning method has received a great deal of attention in recent years, such as InfoGAN (Chen et al., 2016) and β-variational auto-encoder (β-VAE) (Higgins et al., 2017). This learning method transforms the input data into low-dimensional representations that are independent of each other while still retaining the important content. Although it has been actively discussed in the field of computer vision, there are few applications in the field of NLP.

In terms of ensuring model robustness, data augmentation is necessary and essential in machine learning today. With regard to this desirable feature, glyph-aware embedding (i.e., image-based character embedding) allows data augmentation without contextual consideration, such as word dropout (Iyyer et al., 2015) and wildcard training (Shimada et al., 2016). Simple data augmentation based on dropout does not consider the features of the input space. If the NLP method based on glyph-aware embedding is highly interpretive, such as a disentangled representation, an effective data augmentation method can be achieved. This improves not only the robustness of the model but also its interpretability.

In this paper, we propose a general-purpose text classification framework that gives interpretability to data augmentation for image-based glyph-aware character embedding, which has the various advantages mentioned above. The framework consists of two novel methods: (1) glyph-aware disentangled character embedding (GDCE) and (2) semantic sub-character augmentation (SSA). Each method has the following simple but effective features:

- The GDCE is obtained from the variational character encoder (VCE), which is the encoder part of the β-VAE. The VCE takes advantage of the β-VAE to create a low-dimensional representation of the characters, where each dimension follows an independent normal distribution. Therefore, the GDCE provides a disentangled character embedding in which each of the dimensions corresponds to the structure of the sub-character.

- The SSA alters only one dimension of the GDCE, which corresponds with altering some part of the shape of the original character, and can present how the character has changed. In other words, these combinations are equivalent to replacing the sub-character of a character with another readable sub-character.

Our framework improves the interpretability of character embedding by the GDCE, and the SSA provides interpretable data augmentation suitable for the GDCE. We verified the text classification ability of our proposed framework using Japanese text classification tasks. [1]

2 Related work

2.1 Glyph-aware Natural Language Processing

Embedding methods based on character images have been proposed with some excellent success (Chen et al., 2015; Sun et al., 2016; Yu et al., 2017; Sun et al., 2019; Dai and Cai, 2017; Shimada et al., 2016; Liu et al., 2017; Kitada et al., 2018; Ke and Hagiwara, 2017; Aldón Mínguez et al., 2016). These methods are also called glyph-aware embedding as they generate embeddings that take into account the shape of the characters or sub-characters. These image-based methods mainly use convolutional neural networks (CNNs) or convolutional auto-encoders (CAEs) (Masci et al., 2011) for character-embedding learning, and they perform well because of the following advantages: (1) they operate without the cumbersome word segmentation required by some Asian languages, and (2) they can apply additional image-based data augmentation.

2.2 Data Augmentation for Natural Language Processing

For NLP tasks, it is challenging to apply data augmentation methods because of the need to consider the context of the text (Sennrich et al., 2016; Jia and Liang, 2016; Silfverberg et al., 2017; Edunov et al., 2018). Several data augmentation methods that do not require text analysis have been proposed for word embedding (Iyyer et al., 2015; Zhang et al., 2016) and character embedding (Shimada et al., 2016). In particular, Shimada et al. (2016) achieved significant performance improvements by applying dropout (Hinton et al., 2012)-based data augmentation to a type of character embedding called wildcard training (WT). However, these methods have little interpretability of what the data augmentation means in the input text, partly due to the lack of interpretability of the embedding itself. Our proposed SSA is improved WT, and it replaces the sub-character of a character with another readable sub-character.

2.3 Learning Interpretable Character Embeddings

For learning a latent representation that can be interpreted, InfoGAN (Chen et al., 2016) and β-VAE (Higgins et al., 2017) are well known. Unlike InfoGAN, β-VAE is stable while training, requires less assumptions about the data, and relies on only a single hyperparameter β. Because of these advantages, several improved models based on β-VAE have been proposed (e.g., Factor-VAE (Kim and Mnih, 2018), HFVAE (Esmaeili et al., 2019)). Therefore, in this paper, we use β-VAE as a VCE to learn interpretable character embeddings.

3 Methodology

In this paper, we propose a new character-based text classification framework that includes a new character embedding method, consisting of glyph-aware disentangled character embedding (GDCE) and semantic sub-character augmentation (SSA). Figure 1 shows an overview of the proposed text classification framework.

3.1 Glyph-aware Disentangled Character Embedding (GDCE)

We obtain the GDCE using the VCE based on the β-VAE. Since the GDCE provides dimensionally independent features, we expect to solve the prob-

[1] The code required to reproduce the experiments is available on GitHub. https://github.com/IyatomiLab/GDCE-SSA

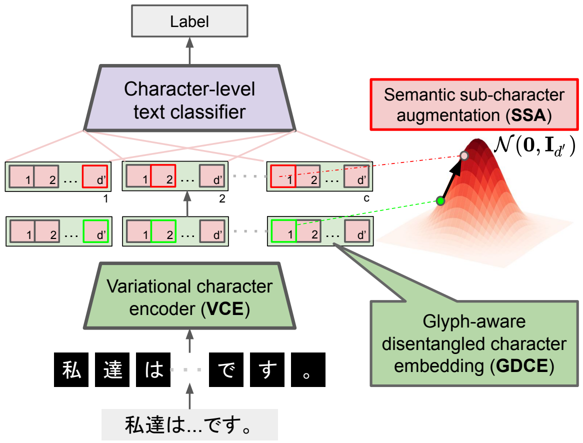

Label
Character-level text classifier
Semantic sub-character augmentation (SSA)
$\mathcal{N}(\mathbf{0}, \mathbf{I}_{d'})$
1 2 … d'
1
2
c
Variational character encoder (VCE)
Glyph-aware disentangled character embedding (GDCE)
私 達 は … で す 。
私達は…です。

Figure 1: Overview of our text classification framework. Each character in the target text is transformed to an image and forwarded as a glyph feature to the subsequent VCE. The VCE is composed of a β-VAE, and it learns the proposed GDCE. Owing to the attractive properties of the GDCE, character-level text classifier can take advantage of the interpretable and highly effective data augmentation method, SSA.

lem of the poorly interpretable character embedding obtained by the CAE.

β-VAE is a generative model that estimates the data distribution $p(\boldsymbol{x})$, where $\boldsymbol{x} \in \mathbb{R}^d$ is a d-dimensional input. Let $\boldsymbol{z} \in \mathbb{R}^{d'}$ be a d'-dimensional latent variable, which is derived from the GDCE in this paper; $p(\boldsymbol{z})$ is a normal distribution, which is the prior distribution of the latent variables, $q(\boldsymbol{z}|\boldsymbol{x})$ is the posterior distribution, and $p(\boldsymbol{x}|\boldsymbol{z})$ is a generative model. We optimize the following function:

$$\begin{aligned}\mathcal{L}_{\beta\text{-VAE}} = \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}[\log p(\boldsymbol{x}|\boldsymbol{z})] \\ - \beta D_{\mathrm{KL}}[q(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z})],\end{aligned} \tag{1}$$

where β is a balancing coefficient for the second term. The first term represents the reconstruction error of the character image. The second term represents the regularization of the latent variables that are learned so as to follow the prior distribution by the KL divergence $D_{\mathrm{KL}}[\cdot||\cdot]$. If the coefficient β increases, it is possible to obtain a representation of the features where each dimension is independent (Higgins et al., 2017).

However, the latent variables themselves are a probability distribution and cannot be backpropagated to the encoder. Hence, the reparameterization trick (Kingma and Welling, 2013) of the approximation method is used. We let $\boldsymbol{\alpha}$ be a sampled random variable from $\mathcal{N}(\mathbf{0}, \boldsymbol{I}_{d'})$ and calculate the latent variables as follows:

$$\boldsymbol{z} = \mu(\boldsymbol{x}) + \boldsymbol{\alpha} \odot \sigma(\boldsymbol{x}), \quad \boldsymbol{\alpha} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_{d'}), \tag{2}$$

where $\odot$ is an element-wise product, μ is the mean of the distribution, and σ is the variance of the distribution. Here, $\mu(\boldsymbol{x})$ and $\sigma(\boldsymbol{x})$ are d'-dimensional vectors obtained from the β-VAE.

3.2 Character-level Text Classification with Semantic Sub-character Augmentation (SSA)

The sequence of c embedded characters $C = \{\boldsymbol{z}^{(1)}, \boldsymbol{z}^{(2)}, \cdots, \boldsymbol{z}^{(c)}\}$ from the GDCE, where $\boldsymbol{z}^{(t)}$ is the t-th character embedding, in the VCE are provided to the following character-level text classifier. The parameters of the classifier are optimized in the back-propagation using the cross-entropy error.

In this paper, we propose SSA as a data augmentation method. Taking advantage of the preferred features of the embedding created by the GDCE, we expect that the sub-character of a character will be replaced by another readable sub-character, using the SSA.

Let γ be the perturbation range, and the formula of the SSA for the i-th dimension $\boldsymbol{z}_i^{(t)}$ of the character embedding $\boldsymbol{z}^{(t)}$ is defined as follows:

$$\boldsymbol{z}_i'^{(t)} = \boldsymbol{z}_i^{(t)} + u, \quad u \sim \mathcal{U}(-\gamma, \gamma), \tag{3}$$

where $u \sim \mathcal{U}(a, b)$ indicates that the random variable u has a uniform distribution with the minimum a and the maximum b. Since each dimension of the GDCE follows $\mathcal{N}(\mathbf{0}, \boldsymbol{I}_{d'})$, the character embedding converted in Eq. 3 falls within the range of trained character-embedding values.

4 Experiment Settings

4.1 Evaluation Datasets

We evaluated our framework with the following datasets: newspaper and livedoor. These datasets were split into two parts: 80% for training and 20% for evaluation. Because these datasets contain new words and/or meanings related to current affairs, accurate word segmentation through morphological analysis has been a challenge in conventional word-level processing for Japanese. Therefore, we can avoid such difficulties by using character-level input instead of word-level input[2].

Newspaper. The newspaper dataset used in Shimada et al. (2016) contains 5,610 Japanese major web newspaper articles (Asahi, Mainichi, Sankei,

[2] It is generally known that a character-level model performs better than a word-level model in Chinese and Japanese (Zhang and LeCun, 2017).

Layer	Encoder
1	Conv2d ($k = (4, 4)$, $o = 32$, $s = 2$) → ReLU
2	Conv2d ($k = (4, 4)$, $o = 32$, $s = 2$) → ReLU
3	Conv2d ($k = (4, 4)$, $o = 64$, $s = 2$) → ReLU
4	Conv2d ($k = (4, 4)$, $o = 64$, $s = 2$) → ReLU
5	Linear($o = 256$) → ReLU
6	Linear($o = 2 \times 10$)

Layer	Decoder
1	Linear ($o = 256$) → ReLU
2	Linear ($o = 1024$) → ReLU
3	Deconv2d ($k = (4, 4)$, $o = 64$, $s = 2$) → ReLU
4	Deconv2d ($k = (4, 4)$, $o = 32$, $s = 2$) → ReLU
5	Deconv2d ($k = (4, 4)$, $o = 32$, $s = 2$) → ReLU
6	Deconv2d ($k = (4, 4)$, $o = 1$, $s = 2$) → Sigmoid

Table 1: Architecture of β-VAE. Kernel size k, output size o, and stride size s was set to the above table.

and Yomiuri) in the categories of politics, the economy, and international news, for a total of 22,440 articles.

Livedoor. The livedoor dataset is commonly used to evaluate models for Japanese.[3] The dataset contains, for example, 870 and 900 Japanese sentences in the categories of movie-enter and sports-watch, respectively. In all the nine categories, it contains a total of 7,367 articles.

4.2 Model Architectures

We trained the VCE based on β-VAE and character-level CNN (CLCNN) (Zhang et al., 2015) as text classifier independently. The hyperparameters of these models were adjusted with a validation set split from the training set, and the predicted results of the evaluation set were reported.

β-variational auto-encoder (β-VAE). Table 1 shows the architecture of β-VAE. Generally, training of β-VAE is unstable, and requires adjustment of hyperparameters. In this paper, we carefully tuned hyperparameters based on Locatello et al. (2019). Adam (Kingma and Ba, 2014) was used to maximize $\mathcal{L}_{\beta\text{-VAE}}$, as shown in Eq. 1. We set train batch size to 64 and the learning rate to 1e-4.

To obtain the GDCE, we trained the VCE with 6,631 common Japanese characters, including Japanese Hiragana, Katakana, and Kanji[4], as well as English alphabets and symbols. These characters were converted to $d = 64 \times 64$ grayscale character images and used as input $\boldsymbol{x}$ to the VCE. We set $\beta = 8$ and $d' = 10$ for all tasks, $\gamma = 1.5$ for the newspaper, and $\gamma = 2.0$ for the livedoor.

Layer	CLCNN
1	Conv1d ($k = 3$, $o = 512$) → ReLU
2	Maxpool1d ($k = 3$, $s = 3$)
3	Conv1d ($k = 3$, $o = 512$) → ReLU
4	Maxpool1d ($k = 3$, $s = 3$)
5	Conv1d ($k = 3$, $o = 512$) → ReLU
6	Conv1d ($k = 3$, $o = 512$) → ReLU
7	Linear (o = #classes)

Table 2: Architecture of CLCNN. Kernel size k, output size o, and stride size s was set to the above table.

Character-level convolutional neural network (CLCNN). Table 2 shows the architecture of CLCNN. We trained CLCNN with the same parameters as in Shimada et al. (2016). Similar to training the character embedding model, Adam was used to minimize the cross-entropy error. We set the learning rate of Adam to 1e-4 and weight decay to 1e-4, train batch size to 256 for the livedoor, and 512 for the newspaper.

In training the CLCNN, we used the GDCE results obtained by the VCE as the input. For training, $c = 128$ consecutive characters were extracted from the text in the newspaper, and $c = 80$ consecutive characters were extracted from the title text in the livedoor. For evaluation, in the newspaper, $c = 128$ characters were slid one by one, the entire text was used as input in the same manner as in Shimada et al. (2016); in the livedoor, it was the same as in the training.

5 Results and Discussion

First, as a comparison of embedding methods, we compared the GDCE with the conventional CAE-based embedding (Shimada et al., 2016). Second, as a comparison of data augmentation methods for image-based character embedding, we also compared the proposed SSA with the conventional WT, the latter of which has reported excellent results but offers no way of interpreting the change on the embedding space.

5.1 Effectiveness of the Proposal on Text Classification

Table 3 presents a comparison of the proposed GDCE and CAE-based embedding. The GDCE showed better document- and sentence-level classification performance than the conventional CAE-

[3] https://www.rondhuit.com/download.html#ldcc

[4] From the Japanese Industrial Standards; first and second levels.

	Accuracy [%]					
	Newspaper			Livedoor		
+ CLCNN	Vanilla	+ WT	+ SSA (**Ours**)	Vanilla	+ WT	+ SSA (**Ours**)
VCE (**Ours**)	**81.02**	82.78	†**84.00**	**67.16**	68.59	†**69.05**
CAE	‡79.81	‡81.62	81.35	‡58.39	‡60.87	60.53

Table 3: A comparison between the VCE (with proposed GDCE) and the CAE in the newspaper and the livedoor results. We compared our proposed framework (presented as †; a disentangled representation) with the state-of-the-art framework of Shimada et al. (2016) (presented as ‡; without the consideration of disentangled representation). Our proposed framework had the highest performance. The model using the VCE performed better than the CAE.

based character embedding without data augmentation. This may be due to the fact that the characters to be learned by the VCE are distributed in a limited embedded space centered on zeros, so the later stage of the CLCNN training became more effective. The WT, which randomly set all representations of a particular character embedding to zero, enhanced the discrimination of both models. The effect on the CAE-based model was particularly large, as reported in previous studies. We can confirm an effect of the WT as a dropout for preventing overfitting, but it did not provide an interpretation of what was changed in the character embeddings.

The proposed SSA provided us with an idea of what the embedding changes would look like, while also providing the same discriminatory capacity as the WT. This may be due to the fact that the GDCE had standardized metrics in the embedding space (i.e., the embedding had a normal distribution), so that the distances between the character embeddings were within the range of what could be assumed. Hence, the size of the perturbations applied could be designed, allowing for meaningful data augmentation. However, the CAE with SSA did not show an improved classification performance. This may be due to the fact that the CAE with SSA does not change to a meaningful character representation.

5.2 Effectiveness of the Proposal on Interpretation

Figure 2 shows a comparison of the reconstructed character images when a $\pm 2.0\sigma$ perturbation is placed on the 2a (the GDCE) and 2b character embedding obtained by the CAE. In Figure 2a, it is confirmed that the shape of the character replaced a different interpretable character or characters with a similar different subcomponent in the input space. In particular, by adding a perturbation to the fifth dimension of the embedding of "迫" or "追" (containing a sub-char. of "辶," meaning *road*), it can be interpreted that it changed to "氵" (sub-char. of *water*) or "⻍" (sub-char. of *road*, the same as "辶"). In addition, by adding a perturbation to the first dimension of the embedding of "綱" or "縄" (containing a sub-char. of "糸," meaning *yarn*), it can be interpreted that it changed to "扌" (sub-char. of *hand*) or "金" (sub-char. of *gold*). From these results, we are convinced that such a replacement in the embedding resulted in more effective data augmentation for training the model.

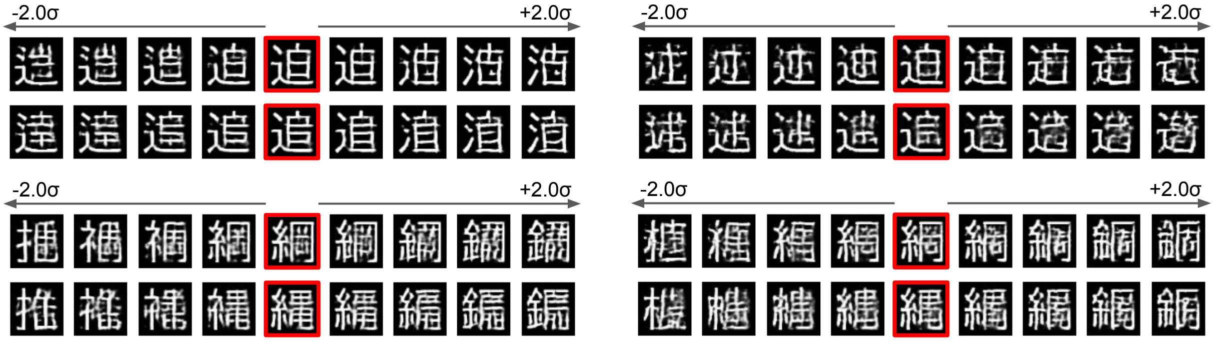
As seen in Figure 2b, in contrast, we were unable to identify these trends. We consider this is one of the typical benefits of our framework in that each dimension of the GDCE is independent and each of them affects each character component (e.g., sub-char. or radical of the character) with independence. In other words, we can change only some part of the character by changing certain dimensions of the embedding.

Since the SSA is a local transformation for the parts of the character shown above, even some characters that do not actually exist are generated by the combination of parts. These are not readable as *correct* characters, but we can make certain interpretations of them. In sum, the combination of the proposed GDCE and SSA provides us with the interpretability of the data augmentation as well as embedding the character while providing a high discriminative power.

5.3 The Effect of Hyperparameters

To understand the effect of hyperparameters, we analyzed the coefficient β and perturbation size γ using the livedoor, as shown in Figure 3.

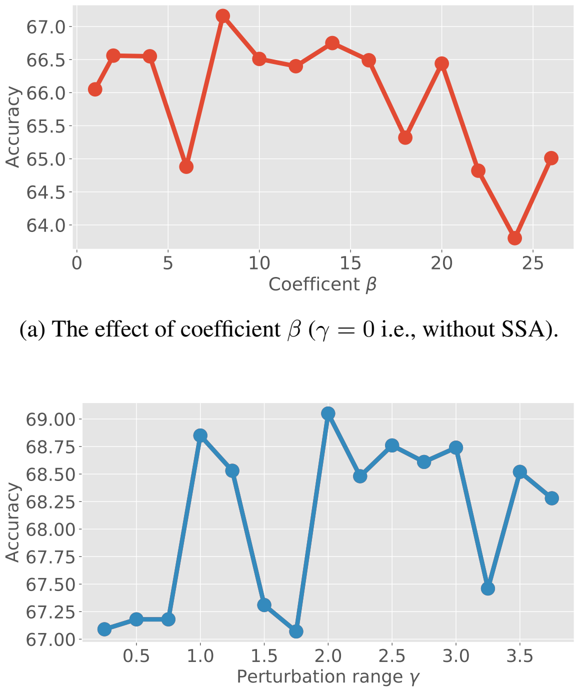
The effect of coefficient β. Figure 3a shows the effect of coefficient β on the evaluation performance with $\gamma = 0$ (i.e., without SSA). In our experiments, we confirmed that $\beta = 8$ is the best from



(a) VCE (**proposed**)

(b) CAE

Figure 2: The results of reconstructing character images from the character embedding trained by the VCE and CAE with perturbation added between 2.0σ. **The upper side** is the reconstructed image of “迫” (approach) and “追” (follow). In the reconstruction from the embedding by the VCE and by adding noise to the fifth dimension of the embedding of “迫” or “追” (containing a sub-char. of “辶” meaning *road*), it can be interpreted that it changed to “氵” (sub-char. of *water*) or “辶” (sub-char. of *road*, the same as “辶”). **The lower side** is the reconstructed image of “綱” (rope) and “縄” (cord). In the reconstruction from the embedding by the VCE and by adding noise to the first dimension of the embedding of “綱” or “縄” (containing a sub-char. of “糸” meaning *yarn*), it can be interpreted that it changed to “扌” (sub-char. of *hand*) or “金” (sub-char. of *gold*).



(a) The effect of coefficient β ($\gamma = 0$ i.e., without SSA).

(b) The effect of perturbation range γ ($\beta = 8$).

Figure 3: The effect of hyperparameters in our framework using the livedoor dataset on the evaluation performance.

the viewpoint of disentanglement and accuracy.

The effect of perturbation size γ. Figure 3b shows the effect of perturbation range γ in SSA on the evaluation performance with $\beta = 8$. Based on the notion that each dimension of the target character embedding follows $\mathcal{N}(\mathbf{0}, \boldsymbol{I}_{d'})$, the perturbation range γ was chosen to be from 1.0σ (covering 68% of the distribution) to 3.0σ (covering almost the entire distribution). The best performance was obtained when the perturbation range was set to $\gamma = 2.0$. This suggests that the character embedding trained by the VCE followed a normal distribution with a mean of $\mu = 0$ and a standard deviation of $\sigma = 1.0$. To cover the distribution, it is considered useful to add perturbation in the range of $\gamma = 2.0$ corresponding to 2.0σ (covering 95% of the distribution).

5.4 Limitations of the Current Study

At present, the role of each dimension in the character reconstruction of the GDCE cannot be clearly defined because it depends on the training of the model. Also, since the VCE was independently trained from the classifier (i.e., not in an end-to-end manner), trained embedding can only consider visual features, not the semantic ones. We will be working on these in the future.

6 Conclusion

We propose a new character-based text classification framework for non-alphabetic languages. As the name implies, the combination of our GDCE and SSA not only provided embedding interpretability but also improved the text classification performance. Our GDCE provided better text classification performance than conventional CAE-based character embedding without data augmentation. Finally, our framework achieved a competitive result to the conventional state-of-the-art CAE-based embedding with WT while also providing model interpretability.

References

David Aldón Mínguez, Marta Ruiz Costa-Jussà, and José Adrián Rodríguez Fonollosa. 2016. Neural machine translation using bitmap fonts. In *Proc. of EAMT HyTra Workshop*, pages 1–9.

Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. 2016. InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. In *Proc. of NIPS*, pages 2172–2180.

Xinxiong Chen, Lei Xu, Zhiyuan Liu, Maosong Sun, and Huanbo Luan. 2015. Joint learning of character and word embeddings. In *Proc. of ICOAI*, pages 1236–1242.

Falcon Z Dai and Zheng Cai. 2017. Glyph-aware Embedding of Chinese Characters. In *Proc. of SCLeM Workshop*, pages 64–69.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding Back-Translation at Scale. In *Proc. of EMNLP*, pages 489–500.

Babak Esmaeili, Hao Wu, Sarthak Jain, Alican Bozkurt, N Siddharth, Brooks Paige, Dana H Brooks, Jennifer Dy, and Jan-Willem Meent. 2019. Structured Disentangled Representations. In *Proc. of AISTAT*, pages 2525–2534.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. 2017. beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. *In Proc. of ICLR*.

Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *CoRR preprint arXiv:1207.0580*.

Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, and Hal Daumé III. 2015. Deep Unordered Composition Rivals Syntactic Methods for Text Classification. In *Proc. of ACL-IJCNLP*, pages 1681–1691.

Robin Jia and Percy Liang. 2016. Data Recombination for Neural Semantic Parsing. In *Proc. of ACL*, pages 12–22.

Yuanzhi Ke and Masafumi Hagiwara. 2017. Radical-level Ideograph Encoder for RNN-based Sentiment Analysis of Chinese and Japanese. In *Proc. of ACML*, pages 561–573.

Hyunjik Kim and Andriy Mnih. 2018. Disentangling by Factorising. In *Proc. of ICML*, pages 2649–2658.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *CoRR preprint arXiv:1412.6980*.

Diederik P Kingma and Max Welling. 2013. Auto-Encoding Variational Bayes. *CoRR preprint arXiv:1312.6114*.

Shunsuke Kitada, Ryunosuke Kotani, and Hitoshi Iyatomi. 2018. End-to-End Text Classification via Image-based Embedding using Character-level Networks. In *Proc. of IEEE AIPR Workshop*, pages 1–4.

Frederick Liu, Han Lu, Chieh Lo, and Graham Neubig. 2017. Learning Character-level Compositionality with Visual Features. *In Proc. of ACL*, pages 2059–2068.

Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. 2019. Challenging Common Assumptions in the Unsupervised Learning of Disentangled Representations. In *Proc. of ICML*, pages 4114–4124.

Jonathan Masci, Ueli Meier, Dan Cireşan, and Jürgen Schmidhuber. 2011. Stacked Convolutional Auto-Encoders for Hierarchical Feature Extraction. In *Proc. of ICANN*, pages 52–59.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving Neural Machine Translation Models with Monolingual Data. In *Proc. of ACL*, pages 86–96.

Daiki Shimada, Ryunosuke Kotani, and Hitoshi Iyatomi. 2016. Document classification through image-based character embedding and wildcard training. In *Proc. of IEEE Big Data Workshop*, pages 3922–3927.

Miikka Silfverberg, Adam Wiemerslage, Ling Liu, and Lingshuang Jack Mao. 2017. Data augmentation for morphological reinflection. In *Proc. of CoNLL-SIGMORPHON*, pages 90–99.

Chi Sun, Xipeng Qiu, and Xuan-Jing Huang. 2019. VCWE: Visual Character-Enhanced Word Embeddings. In *Proc. NAACL-HLT*, pages 2710–2719.

Fei Sun, Jiafeng Guo, Yanyan Lan, Jun Xu, and Xueqi Cheng. 2016. Inside out: Two jointly predictive models for word representations and phrase representations. In *Proc. of AAAI*.

Jinxing Yu, Xun Jian, Hao Xin, and Yangqiu Song. 2017. Joint Embeddings of Chinese Words, Characters, and Fine-grained Subcharacter Components. In *Proc. of EMNLP*, pages 286–291.

Dongxu Zhang, Tianyi Luo, and Dong Wang. 2016. Learning from LDA Using Deep Neural Networks. In *Proc. of NLPCC-ICCPOL*, pages 657–664.

Xiang Zhang and Yann LeCun. 2017. Which Encoding is the Best for Text Classification in Chinese, English, Japanese and Korean? *CoRR preprint arXiv:1708.02657*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Proc. of NIPS*, pages 649–657.