Experiments on Unsupervised Text Classification based on Graph Neural Networks

Haoxiang SHI†, Cen WANG††, and Tetsuya SAKAI†

† Faculty of Science and Engineering, Waseda University
3-4-1 Okubo, Shinjuku-ku, Tokyo, 169-8555, Japan
†† KDDI Research Inc.,
2-1-15 Ohara, Fujimino-shi, Saitama-ken, 356-0003, Japan
E-mail: †hollis.shi@toki.waseda.jp, ††ce-wang@kddi-research.jp, †††tetsuya@waseda.jp

Abstract Traditional text classification requires labelled data for training the models, but obtaining a sufficient amount of training data is costly and often not practical. Recently, an unsupervised approach to text classification has been proposed, which is based on a novel neural network architecture that represents textual relationships by graphs. The present study reports on an experiment to validate this approach: our model achieves 42.5% in average classification accuracy and outperforms a basic autoencoder baseline with an accuracy of 31.3%.
Key words Text Classification, Graph Neural Network, Unsupervised Deep Learning.

1 Introduction

Text classification is a critical language task that can support multiple applications, such as fake news identification, spam detection, sentiment analysis and opinion mining [1]. Recently, supervised deep learning methods have made great progress in text classification tasks. The majority of supervised deep learning methods focus on the design of textual encoder [2][3]. After proper encoding, classification task can obtain high accuracy [4]. These procedures may require mass labeled data (i.e. corpus) to train the deep learning model. However, data in real world is often unlabeled. To label data (i.e. to build a corpus) is costly and inefficient. In order to make natural language processing (NLP) models versatile, the effectiveness of unsupervised models should be explored.

Traditionally unsupervised text classification models use word frequency or spare word vectors to represent text, and then text classification is done according to the features lying in the corresponding representations. These simple representation approaches cannot preserve context. As a result, the text classification performance may deteriorate. To further promote effectiveness, it is necessary to find a contextualization representation without a supervised pre-training encoder.

Auto-encoder (AE) neural networks [5] can generate a powerful automatic representation in an unsupervised manner. On the other hand, graph representation is a better solution to realize contextualization [6]. A common way to convert a text into a graph is to use co-occurrence window (e.g. TextRank [7]). Graph auto-encoder (GAE) neural network is an advanced approach that combines the advantage of AE and graph representation. Nevertheless, GAE can only encode text graph into few dimensions, which may cause insufficient text feature representation to pursue accurate classification. To deal with such a problem, inspired by a task of citation link prediction [8], variational graph autoencoder (VGAE) neural network has been applied to text classification. Unlike conventional graph auto-encoder, VGAE provides a probabilistic manner for representing an observation in the latent space of a text. Thus, the encoder can be formulated to represent a probability distribution for each latent attribute of a text. Upon high-dimensional latent space representation, more accurate unsupervised text classification could be achieved.

In this paper, using the 20newsgroup dataset [9], we have evaluated the effectiveness of VGAE in a text classification experiment. The results suggest that VGAE outperforms state-of-the-art unsupervised deep learning methods, namely, AE and GAE. The code of our work can be found in https://github.com/wcdtom/Underan/.

2 Related Work

Unsupervised text classification has a long history. The basic idea to classify text in an unsupervised manner, in fact, is to cluster texts with similar features. In the early years,

simple features such as word frequency (e.g. TF-IDF [10]) and topic words (e.g. TextRank) have been used.

With the development of deep learning, text can be represented as deep latent features. The state-of-the-art neural network architectures composed by convolutional neural networks (CNNs) or long-short term memory (LSTM) cells cannot represent latent features with semantic relativity unless there is training data. It is because CNN and LSTM cannot naturally support relativity representations. To this end, in supervised learning, graph-based text representation and classification methods have been proposed in recent years. Fu et al. [6] have proposed the GraphRel that uses graph convolutional neural network (GCN) to realize relation extraction. Then, Yao et al. [1] have implemented text classification based on GCN, and achieved the highest accuracy when comparing to CNN and LSTM -based approach.

In unsupervised learning, graph-based text representation and classification have also received much attention. Several studies focus on building more complex graph to describe text. For example, they add special edges to indicate grammatical relation between two vertexes (i.e. words) [11]. Then, they use unsupervised text classification to verify the modified graph representation. Hinton et al. [5] have proposed the epoch-making AE, which enables the unsupervised working manner of the neural network. Kingma et al. [12] proposed the variational autoencoder (VAE) neural network to enable more generative representations. Meanwhile, based on graph-like dataset and utilizing powerful GCN from supervised models, Kipf et al. [13] proposed GAE to do the semi-supervised classification task. Following that, Wang et al. [14] have compared unsupervised text classification under multiple kinds of auto-encoder neural networks. The contrastive results shows the limitation of the models of AE, VAE and GAE. Further, Kipf et al. [6] proposed VGAE which combines the advantages of GAE and VAE methods. VGAE has been successfully utilized on citation link prediction. Inspired by Kipf's work, we want to investigate whether the VGAE performs well in other language tasks. Specifically, the present study considers the task of text classification.

3 Text Classification via VGAE

VGAE inherits the architecture of AE, which includes encoder and decoder parts. The encoder part represents the inputs into latent features. And decoder part uses the latent features to generate the outputs, which can be regarded as the regeneration of the inputs. The learning feedback can be based on the difference between inputs and outputs.

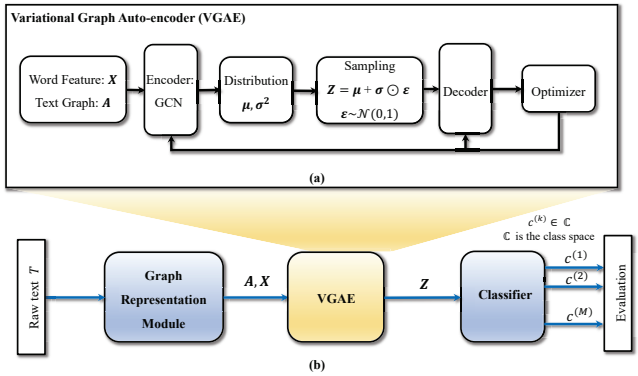

Fig. 1 Text classification by means of VGAE

Figure 1 Text classification by means of VGAE

As aforementioned, basic AE can only represent an input into a single point (i.e. usually 2D or 3D). Hence, VAE adds latent variables to the AE. The main idea of "variational" is to restrict the learned parameters from a known distribution. For any sampling of the latent distributions, the decoder is expected to accurately reconstruct the input. The statistical distribution helps to force a continuous, smooth latent space representation. Namely, the values in distributions are near to one another in latent space, which could contribute a more similar reconstruction.

While GAE is a variant of AE, the input of GAE is a graph. The encoder of the GAE is GCN. Using the encoded features of the GCN, output graph can be regenerated via decoder. The procedures of GAE can be described as follows:

$$\mathbf{Z} = GCN(\mathbf{A}, \mathbf{X})$$
$$\hat{\mathbf{A}} = sigmoid(\mathbf{Z}\mathbf{Z}^{\top})$$

where, $\mathbf{A}$ is the input graph, $\mathbf{X}$ is the feature matrix of the nodes in graph $\mathbf{A}$ (the specific form of $\mathbf{X}$ will be detailed later). The $\mathbf{Z}$ is the representations via the encoder. $\hat{\mathbf{A}}$ is a regeneration of graph $\mathbf{A}$. The learning approaches (i.e. the loss function) of a GAE model can be the cross entropy.

VGAE is a combination of GAE and VAE. The calculation of VGAE is depicted in Fig .1(a). The encoder of VGAE is as follows:

$$\mu = GCN_{\mu}(\mathbf{A}, \mathbf{X})$$
$$\log \sigma = GCN_{\sigma}(\mathbf{A}, \mathbf{X})$$
$$q(\mathbf{Z}|\mathbf{A}, \mathbf{X}) = \prod_{i=1}^{N} q(\mathbf{z}_i|\mathbf{A}, \mathbf{X})$$
$$q(\mathbf{z}_i|\mathbf{A}, \mathbf{X}) = \mathcal{N}(\mathbf{z}_i|\mu_i, diag(\sigma_i^2))$$

where, μ is the set of each matrix μ_i, σ is the set of each matrix σ_i. A two-layer GCN is defined as:

$$GCN(\mathbf{A}, \mathbf{X}) = \tilde{\mathbf{A}} Relu(\tilde{\mathbf{A}}\mathbf{X}\mathbf{W}_0)\mathbf{W}_1$$

where, $\tilde{\mathbf{A}} = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{1/2}$ is the symmetrically normalized adjacent matrix, and $Relu(\cdot) = max(0, \cdot)$.

The decoder of VGAE is given as:

$$p(\mathbf{A}|\mathbf{X}) = \prod_{i=1}^{N}\prod_{j=1}^{N} p(a_{ij}|\mathbf{z}_i, \mathbf{z}_j)$$
$$p(a_{ij} = 1|\mathbf{z}_i, \mathbf{z}_j) = sigmoid(\mathbf{z}_i^\top \mathbf{z}_j)$$

where, a_{ij} is an element of the graph $\mathbf{A}$.

The loss function of the VGAE is:

$$\mathcal{L} = \mathbb{E}_{q(\mathbf{Z}|\mathbf{A},\mathbf{X})}[\log p(\mathbf{A}|\mathbf{Z}] - KL[q(\mathbf{Z}|\mathbf{A}, \mathbf{X})||p(\mathbf{Z})]$$

The loss function during learning consists of two parts: constructing loss and the latent variable restriction loss. Constructing loss is to measure to what extent the constructed adjacency matrix is similar to the input one; the other loss is to apply KL-divergence [15] to measure how similar the distribution of the latent variable and a normal distribution are.

Upon the VGAE, the procedure of the unsupervised text classification is described in Fig. 1(b). Texts are firstly embedded into graphs. In a single text graph $\mathbf{A}$, words and texts are regarded as a node, and the node feature is a one-hot vector. All the vectors form the text feature matrix $\mathbf{X}$. Matrices $\mathbf{A}$ and $\mathbf{X}$ are inputted into VGAE, then the latent feature $\mathbf{Z}$ is retrieved by the classifier. Text classification is to cluster all the feature matrices. Thus, classifier can be implemented by any clustering method, such as k-means, SVM and spectral clustering [16].

4 Experimental Results

In order to evaluate the effectiveness of VGAE in text classification, we take 18,821 texts in the 20newsgroup dataset as inputs of experiment. After the VGAE training, 16 dimensions of latent feature are obtained. Then the k-means method is selected to classify the texts according to the latent features.

We use classification accuracy as our evaluation measure, which is defined as:

$$ACC = \frac{\sum_{i=1}^{n} \delta(c_{t_i}, map(\hat{c}_{t_i}))}{n}$$

where c_{t_i} is the real label of text i,$i = 1, 2, \ldots, n$, and $\hat{c}_{t_i}$ is the predictive label of text i. If $c_{t_i} = map(\hat{c}_{t_i})$, then $\delta(c_{t_i}, map(\hat{c}_{t_i})) = 1$. The function $map(\cdot)$ indicates a permutation mapping that best matches the predictive clustering labels to real labels.

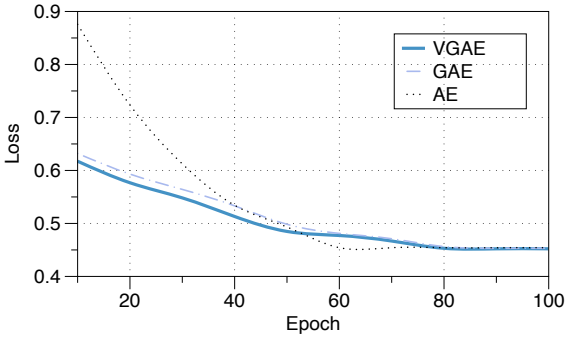

Figure 2 The training loss of each unsupervised model

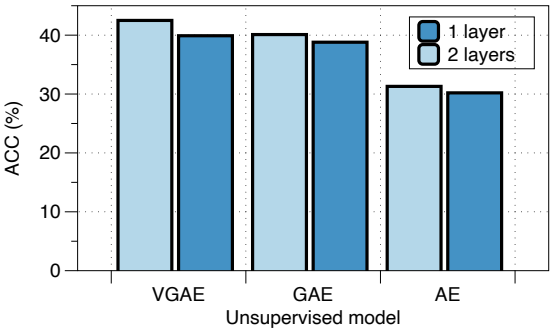

Figure 3 The ACC of VGAE, GAE, AE with one-layer and two-layer

The training loss (with one-layer model) of the AE, GAE and VGAE are depicted in Fig. 2. It can be found that the training loss of AE reduced very fast. It may be because that the architecture of the model is simple. On the other hand, the GAE and VGAE converge slowly, which is caused by the more complex inner structure of graph neural network.

The ACCs of AE, GAE and VGAE are shown in Fig. 3. Both one-layer model and two-layer model are evaluated. The results suggest that the two-layer VGAE can achieve the highest ACC, and VAE is the next, and both VGAE and VAE are more effective than the basic AE for unsupervised text classification.

5 Conclusion

In this paper, we have verified the effectiveness of VGAE in text classification empirically. In the experiment, we compared VGAE to other two methods, GAE and AE. In terms of accuracy in text classification, VGAE outperforms GAE, which in turn outperforms AE. In addition, two-layer networks outperform their one-layer counterparts for every model.

For future work, we would like to use other datasets to evaluate the VGAE text classification model. Additionally, the fine-tuning on this model can be further investigated.

Acknowledgement

We would like to thank Mr. Zhaohao Zeng (Ph.D Candidate) for his valuable advice.

References

[1] Yao Liang, Chengsheng Mao and Yuan Luo. "Graph convolutional networks for text classification", Proc. of the AAAI, vol. 33, 2019.

[2] K. S. Tai, R. Socher and C. D. Manning, "Improved semantic representations from tree-structured long short-term memory networks", Proc. of ACL, pp. 1556–1566, 2015.

[3] J. Pennington, R. Socher and C. Manning, "Glove: Global vectors for word representation", Proc. of the EMNLP, pp. 1532–1543, 2014.

[4] G. Wang, C. Li, et al. "Joint embedding of words and labels for text classification", Proc. of the ACL, pp. 2321–2331, 2018.

[5] E. Geoffrey Hinton and Salakhutdinov Ruslan "Reducing the dimensionality of data with neural networks", Science vol. 313 5786, pp. 504-507, 2006.

[6] Tsu-Jui Fu, Peng-Hsuan Li and Wei-Yun Ma. "GraphRel: Modeling text as relational graphs for joint entity and relation extraction", Proc. of the ACL, 2019.

[7] R. Mihalcea and P. Tarau, "TextRank: Bringing order into texts", Proc. of ACL, pp. 404-411, 2004.

[8] Thomas N. Kipf, and Max Welling, "Variational Graph Auto-Encoders", in Bayesian Deep Learning Workshop of the NIPS, 2016.

[9] [Online Available.] 20newsgroup: http://qwone.com/ jason/20Newsgroups/

[10] C. S. Saranyamol and L. Sindhu. "A survey on automatic text summarization", Int. J. Comput. Sci. Inf. Technol, vol. 5, no. 6, pp. 7889-7893, 2014.

[11] Esteban Castillo, et al. "Analysis Using Different Graph-Based Representations", Computacion y Sistemas, Vol. 21, No. 4, pp. 581–599, 2017.

[12] [Online Available] P. Diederik P. and Welling Max, "Auto-Encoding Variational Bayes": https://arxiv.org/abs/1312.6114.

[13] T. N. Kipf and M. Welling. "Semi-supervised classification with graph convolutional networks", Proc. of ICLR, pp. 1-14, 2017.

[14] Shiping Wang, et al. "An Overview of Unsupervised Deep Feature Representation for Text Categorization", IEEE Transactions on Computational Social Systems, vol. 6, no. 3, pp. 504-517, 2019.

[15] Severin Klingler, et al. "Efficient Feature Embeddings for Student Classification with Variational Auto-Encoders", Proc. of the 10th International Conference on Educational Data Mining, pp.73-79, 2017.

[16] [Online Available] KL-divergence: https://en.wikipedia.org/wiki/Kullback%E2%80%93... ...Leibler_divergence.