Unsupervised Neural Segmentation and Clustering for Unit Discovery in Sequential Data

Jan Chorowski
University of Wroclaw

Nanxin Chen
John Hopkins University

Ricard Marxer
Université de Toulon

Hans J.G.A. Dolfing

Adrian Łańcucki
University of Wroclaw

Guillaume Sanchez
Université de Toulon

Sameer Khurana
Massachusetts Institute of Technology

Tanel Alumäe
Tallinn University of Technology

Antoine Laurent
Le Mans University

Abstract

We study the problem of unsupervised segmentation and clustering of handwritten lines with applications to character discovery. We propose a constrained variant of Vector Quantized Variational Autoencoder (VQ-VAE) which produces a discrete and piecewise-constant encoding of the data. We show that the constrained quantization task is dual to a Markovian dynamics prior placed on the latent codes. Such view facilitates a probabilistic interpretation of the constraints and allows efficient inference. We demonstrate the effectiveness of the proposed method in the context of unsupervised handwriting character discovery in 17th-century scanned manuscripts.

1 Introduction

Aligning a fixed-rate stream of observations to a variable-rate series of categories is a problem that often arises with sequential data. In speech processing, equal-sized chunks of audio are mapped to phones of different duration. Similarly, in scanned handwritten documents, contiguous ranges of pixel columns in an image are assigned to characters. The problem arises even in non language-related signals like segmenting and categorizing mouse behavior from video recordings [1]. The problem becomes harder in an unsupervised setting, where learning of the alignment has to take place jointly with learning of the underlying latent representations.

In this study we show how to enforce the prior knowledge about average character duration in a way that is easy to implement and behaves well during training. Using deep learning approaches such as discrete bottleneck variational autoencoders [2, 3], we demonstrate how to apply a Markovian prior over discrete latent codes by enforcing segmentation in a VQ-VAE with a PixelCNN decoder. The proposed solution is tested on manuscript data. The experimental results show that it improves the agreement of learned representations with ground truth alignments, measured by training a week classifier on the discovered, latent representation.

In addition, we show that the HMM prior can be applied using an equivalent dual constraint based approach. The two formulations bring complementary benefits. The former facilitates efficient application during inference. The latter is advantageous during training because it requires a single

Workshop on Perception as Generative Reasoning, NeurIPS 2019, Vancouver, Canada.

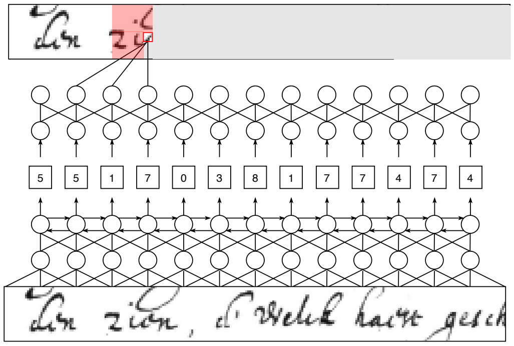

Figure 1: Model architecture. The encoder combines convolutional and recurrent layers and extracts a sequence of embeddings, which are quantized and used as conditioning of an autoregressive PixelCNN decoder. The decoder produces consistent samples, by filling in the details missing in the low-bandwith encoding with information form the autoregressive path.

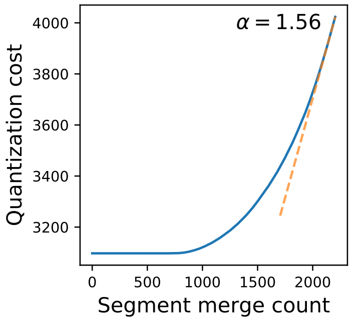

Figure 2: The Lagrange multiplier α of a constraint corresponds to increase of the objective function for an infinitesimal change in the constraint.

intuitive hyperparameter and does not tend to under-segment the data, thus mitigating the problem of latent variable collapse [4].

2 Model Description

Our model contains an encoder, followed by a VQ-VAE quantizer [3] and PixelCNN [5] decoder, which reconstructs its input in an unsupervised fashion (Figure 1). It processes lines of handwritten text as grayscale images $x \in [0,1]^{H\times L}$ of fixed height H and variable length L. The encoder, inspired from temporal architectures for ASR [6], is a stack of convolutional and recurrent layers which reduce the length of the input sequence by a constant factor k to $L' = \lceil L/k \rceil$, producing a sequence of *prequantized* encodings $e(x) \in \mathbb{R}^{D\times L'}$.

Encodings are then quantized by replacing each vector in e by one of N learned prototypes: $q_t = \text{quantize}(e_t) = E_{z_t}$, where $E \in \mathbb{R}^{D\times N}$ is a matrix of prototypes and $z \in \{0, 1, \dots N-1\}^{L'}$ is the sequence of the respective prototype IDs. Finally, the sequence q is upsampled k times and used as conditioning of a PixelCNN decoder that regenerates the input pixel-by-pixel in a left-to-right and top-to-bottom order. The model is trained to minimize a sum of 3 terms, consisting of the reconstruction log-likelihood, the distance of the prequantized encoding to the prototype, which is backpropagated only to prototypes, and the commitment loss, which forces the encoder to produce representations which are close to the quantized values:

$$\mathcal{L} = -\log p(x|q(x)) + ||\text{sg}(e(x)) - q(x)||^2 + \gamma||e(x) - \text{sg}(q(x))||^2 \tag{1}$$

where sg denotes the stop-gradient operation. The straight-through estimator [3, 7] is used to backpropagate the loss derivative through the quantization operation.

2.1 Enforcing Segmentation

The original VQ-VAE [3] quantizes all encodings separately, disregarding their order. However, desired latent categories (i.e. characters) span contiguous, variable width segments of encoding vectors. Thus, the quantizer should assign all vectors in a segment to the same prototype. Therefore all L' vectors $e(x)$ need to be quantized jointly in a way which assigns neighboring vectors to the same prototype.

To this end we introduce a constrained optimization problem which is forced to create S segments:

$$\min_{z_1, z_2, \dots, z_{L'}} \sum_{l=1}^{L'} ||e(x)_l - E_{z_l}||^2 \quad \text{s.t.} \quad \sum_{t=2}^{L'} [z_t \neq z_{t-1}] = S - 1, \tag{2}$$

where $[\cdot]$ is the indicator function. While it can be solved exactly using a dynamic algorithm with a runtime $O(NLS)$, we approximately solve it in $O(NL \log L)$ using greedy merging of neighboring codes until obtaining the desired number of segments.

The number of characters in a line varies considerably with scribe, and even in-and-between lines of the same scribe, typically being more narrow towards ends of lines and pages. The data also contains long empty spaces at paragraph ends which are not automatically truncated. The constraint in (2) thus cannot be enforced with the same value of parameter S to each line. Instead, we enforce it during training across whole minibatches, on which the actual number of characters tends to be closer to the expected number computed based on mean character density.

During testing we must be able to apply the model independently to each scanned line. To this end, in Appendix A we obtain an equivalent dual form of (2):

$$\min_{z_1, z_2, \ldots, z_{L'}} \sum_{l=1}^{L'} ||e(x)_l - E_{z_l}||^2 + \sum_{t=2}^{L'} \alpha' \log p_\rho(z_l | z_{l-1}), \tag{3}$$

where α' is an appropriate constant, and p_ρ defines a prior transition probability of the latent codes:

$$p_\rho(z_l | z_{l-1}) = \begin{cases} \rho & \text{if } z_l = z_{l-1}, \\ \frac{1-\rho}{N} & \text{otherwise.} \end{cases} \tag{4}$$

The prior states that with probability ρ latents z_l and z_{l-1} are part of the same segment, and with probability $1 - \rho$ there is a segment boundary between columns $l - 1$ and l. The model is equally likely to assign each of N characters to the new segment.

The unconstrained problem (3) corresponds to finding a quantization under a simple Markovian prior placed on the latent codes z. It can be efficiently solved in $O(NL)$ using the Viterbi algorithm. Moreover, the batch-wise enforced constraint is removed allowing independent test line processing.

We can see that the two formulations (2) and (3) have their own advantages. During training, the constrained form uses an easy to set and intepret hyperparamter S that corresponds to the desired number of segments in a minibatch. Furthermore, it decouples the HMM cost, related to the segment duration, from the quantization error, dependent among others on the magnitude of $e(x)$. Many factors of the encoder design will affect these values, such as initial weight scaling and presence of normalization layers [8]. In fact, using (2) was crucial for using recurrent layers in the encoder – we have observed that during training the magnitude of their outputs changes considerably. Moreover, the constraint forces the network to find a given number of segments avoiding the trivial solution in which a single segment is created, all latent codes became equal, and the decoder learns to ignore the latent variables [4]. On the other hand, the HMM formulation (3) has a direct probabilistic interpretation and enables an efficient and correct application of the model to new data.

Finally, we observe that segmental quantization provides additional training signal to the encoder. Minimization of the commitment term $||e(x) - \mathrm{sg}(q(x))||^2$ in (1) forces the encoder to produce a representation close to the quantized prototype. The encoder is therefore trained to produce the same output for all columns inside segments and sharp output changes between them.

3 Experimental Results

We perform the experiments on a historical corpus of 17th century Early Modern Dutch concerning ship journals of the Dutch East India Company[1]. The encoder is a stack of 7 convolutions, followed by 2-layer BiLSTM network with hidden size 32. The VQ-VAE bottleneck has 256 tokens of size 64. The decoder is composed of 2 dilated convolutions, followed by 2 regular convolutions, and two 7-layer PixelCNNs, one reconstructing the image left-to-right and one working in the opposite direction. We train the model using Adam with learning rate 2×10^{-3}, multiplying it by 0.8 whenever validation loss plateaus twice in a row.

We show the segmentation learned by the model in Figure 3. The discovered segment boundaries (shown in the bottom half of the picture) approximately match the ground truth ones (shown in the top half of the picture) which were extracted out of a CTC system trained in a supervised way.

[1] The corpus is in preparation for publishing

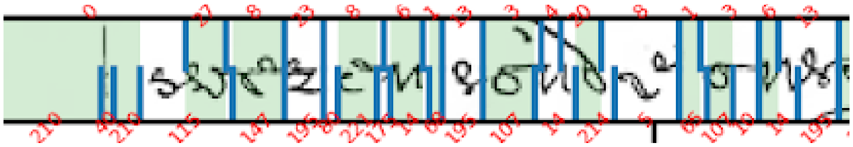

Figure 3: Unsupervised segmentations obtained with the proposed criterion.
The top half shows ground truth segmentation (vertical bars) and character codes. The bottom half shows model segmentation and prototype indices (out of 256). Green-shaded columns are properly classified by a many-to-one mapping of prototype to characters. For instance, we can see the model learned to properly delineate and recognize the character "o", consistently assigning it to cluster 107.

Table 1: Evaluation of the learned representation on downstream tasks and clustering quality metrics.

Model	Downstream CER	Prototype to Char Acc.	ARI	AMI
Baseline VQ-VAE	48%	34%	0.15	0.15
Segmental VQ-VAE	29%	58%	0.24	0.36

When discovered units are matched to characters, the system correctly classifies more than half of columns (marked in green), for example all visible occurrences of letter "o" are correctly identified as belonging to a single category. In Table 1 we show that enforcing segments improves the quality of the representation for downstream supervised tasks. We also show that the segmentation predicted using the proposed method is in higher agreement with the ground truth alignment via clustering metrics Adjusted Rand Index (ARI) [9] and Adjusted Mutual Information (AMI) [10].

4 Related Work

Design of flexible priors for variational autoencoders is an active research area. In [1] probabilistic graphical models are used to specify the prior. This approach has been applied to unit discovery in speech using an HMM-VAE model [11, 12]. We also apply a Markovian prior, albeit specified implicitly through a dual constraint optimization problem.

Classical unsupervised unit discovery methods rely on priors that specify unit-dependent HMM models [13, 14] with simple GMM emission models. Instead, we use a simple character structure, and rely on deep learning models for image generation.

Slow feature analysis [15] has been employed to find slowly changing latent representations. However, straightforward application of a penalty on the magnitude of latent encoding changes results in posterior collapse of a VAE[16]. Instead, our segmental criterion prevents the collapse and trains the encoder to produce a piece-wise constant representation with infrequent, yet abrupt changes.

5 Conclusion

We have demonstrated an alternative, constraint-based way to enforce an HMM prior over latent variables of a variational autoencoder. An important benefit of the constraint-based formulation is its ease of use. It is based on an intuitive, easy to set hyperparameter which controls the average segment length. Moreover, it does not extend the training cost with a term which leads training to poor solutions producing constant latent representations. The benefit of the approach is shown with improvements in unsupervised detection of characters in historical handwriting. We believe, however, that the highlighted duality between probabilistic priors and constraints has a broader set of applications and will ease specifying prior assumptions about the latent space.

Acknowledgements

The research reported here was conducted at the 2019 Frederick Jelinek Memorial Summer Workshop on Speech and Language Technologies, hosted at L'École de Technologie Supérieure (Montreal, Canada) and sponsored by Johns Hopkins University with unrestricted gifts from Amazon, Facebook, Google, and Microsoft. Authors also thank Polish National Science Center for funding under the Sonata 2014/15/D/ST6/04402 grant the PLGrid project for computational resources.

References

[1] Matthew Johnson, David K Duvenaud, Alex Wiltschko, Ryan P Adams, and Sandeep R Datta. Composing graphical models with neural networks for structured representations and fast inference. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 2946–2954. Curran Associates, Inc., 2016.

[2] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. URL `http://arxiv.org/abs/1312.6114`.

[3] Aaron van den Oord, Oriol Vinyals, and koray kavukcuoglu. Neural discrete representation learning. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6306–6315. Curran Associates, Inc., 2017. URL `http://papers.nips.cc/paper/7210-neural-discrete-representation-learning.pdf`.

[4] Samuel Bowman, Luke Vilnis, Oriol Vinyals, Andrew M Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. In *Proceedings of the Twentieth Conference on Computational Natural Language Learning (CoNLL).*, 2016.

[5] Aaron Van Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1747–1756, New York, New York, USA, 20–22 Jun 2016. PMLR. URL `http://proceedings.mlr.press/v48/oord16.html`.

[6] Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang Chen, et al. Deep speech 2: End-to-end speech recognition in english and mandarin. In *International conference on machine learning*, pages 173–182, 2016.

[7] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

[8] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pages 448–456. JMLR.org, 2015. URL `http://dl.acm.org/citation.cfm?id=3045118.3045167`.

[9] Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of classification*, 2(1): 193–218, 1985.

[10] Nguyen Xuan Vinh, Julien Epps, and James Bailey. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *Journal of Machine Learning Research*, 11(Oct):2837–2854, 2010.

[11] Janek Ebbers, Jahn Heymann, Lukas Drude, Thomas Glarner, Reinhold Haeb-Umbach, and Bhiksha Raj. Hidden Markov Model Variational Autoencoder for Acoustic Unit Discovery. In *Interspeech 2017*, pages 488–492. ISCA, August 2017. doi: 10.21437/Interspeech.2017-1160.

[12] Thomas Glarner, Patrick Hanebrink, Janek Ebbers, and Reinhold Haeb-Umbach. Full Bayesian Hidden Markov Model Variational Autoencoder for Acoustic Unit Discovery. In *Interspeech 2018*, pages 2688–2692. ISCA, September 2018. doi: 10.21437/Interspeech.2018-2148.

[13] Chia-ying Lee and James Glass. A Nonparametric Bayesian Approach to Acoustic Model Discovery. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 40–49, Jeju Island, Korea, July 2012. Association for Computational Linguistics.

[14] Lucas Ondel, Lukaš Burget, and Jan Černocký. Variational Inference for Acoustic Unit Discovery. *Procedia Computer Science*, 81:80–86, 2016. ISSN 18770509. doi: 10.1016/j.procs.2016.04.033.

[15] Laurenz Wiskott and Terrence J. Sejnowski. Slow Feature Analysis: Unsupervised Learning of Invariances. *Neural Computation*, 14(4):715–770, April 2002. ISSN 0899-7667. doi: 10.1162/089976602317318938.

[16] Jan Chorowski, Ron J. Weiss, Samy Bengio, and Aäron van den Oord. Unsupervised speech representation learning using WaveNet autoencoders. *arXiv:1901.08810 [cs, eess, stat]*, January 2019.

A Derivation of the dual loss

During training the model chooses a quantized representation that creates exactly S segments and minimizes

$$\min_{z_1,z_2,\ldots,z_{L'}} \sum_{l=1}^{L'} ||e(x)_l - E_{z_l}||^2 \quad \text{s.t.} \sum_{t=2}^{L'} [z_t \neq z_{t-1}] = S - 1, \tag{5}$$

To obtain the dual, unconstrained form we first transform the constrained quantization criterion (5) into an unconstrained one by taking its Lagrangian function:

$$\sum_{l=1}^{L'} ||e(x)_l - E_{z_l}||^2 + \alpha \left(\sum_{t=2}^{L'} [z_t \neq z_{t-1}] - S - 1 \right), \tag{6}$$

where α is a Lagrange multiplier. At optimum, α equals the objective increase for an infinitesimal change of the constraint (Figure 2). We approximate α during model training by tracking the mean increase of the objective (5) over the last 20 segment merges performed by the greedy algorithm.

Finally, we can gain further insights into the structure of the problem by introducing a constant

$$\alpha' = \alpha / \log \frac{1-\rho}{N\rho},$$

where N is the number of symbols and ρ is an assumed a-priori probability of continuing the previous segment. Observe that

$$\alpha \left(\sum_{t=2}^{L'} [z_t \neq z_{t-1}] - S - 1 \right) = \tag{7}$$

$$= \alpha \sum_{t=2}^{L'} [z_t \neq z_{t-1}] + C_1 = \tag{8}$$

$$= \alpha' \sum_{t=2}^{L'} [z_t \neq z_{t-1}] \log \frac{1-\rho}{N\rho} + C_1 = \tag{9}$$

$$= \alpha' \sum_{t=2}^{L'} [z_t \neq z_{t-1}] \log \frac{1-\rho}{N\rho} + \alpha' \sum_{t=2}^{L'} \left([z_t \neq z_{t-1}] + [z_t = z_{t-1}] \right) \log \rho + C_2 = \tag{10}$$

$$= \alpha' \sum_{t=2}^{L'} [z_t \neq z_{t-1}] \log \frac{1-\rho}{N} + \alpha' \sum_{t=2}^{L'} [z_t = z_{t-1}] \log \rho + C_2 = \tag{11}$$

$$= \alpha' \sum_{t=2}^{L'} \log p_\rho(z_l | z_{l-1}) + C_2, \tag{12}$$

where we use the fact that $[z_t \neq z_{t-1}] + [z_t = z_{t-1}] = 1$ and p_ρ is defined to be:

$$p_\rho(z_l|z_{l-1}) = \begin{cases} \rho & \text{if } z_l = z_{l-1}, \\ \frac{1-\rho}{N} & \text{otherwise.} \end{cases} \tag{13}$$

Since adding a constant to an optimization target doesn't change the location of the optimum, finding the minimizer of (6) is equivalent to minimizing

$$\min_{z_1,z_2,\ldots,z_{L'}} \sum_{l=1}^{L'} ||e(x)_l - E_{z_l}||^2 + \sum_{t=2}^{L'} \alpha' \log p_\rho(z_l|z_{l-1}) \tag{14}$$

The prior $P_\rho(\cdot)$ states that with probability ρ latents z_l and z_{l-1} are part of the same segment, and with probability $1-\rho$ there is a segment boundary between columns $l-1$ and l. The model is equally likely to assign each of N characters to the new segment.