Context-Dependent Robust Text Recognition using Large-scale Restricted Bayesian Network

Hidemoto Nakada and Yuuji Ichisugi

Abstract We have been proposing a computational model of the cerebral cortex called BESOM, which models the cerebral cortex as restricted Bayesian networks based on recent findings in the neuroscience area. Since BESOM is based on Bayesian network, it inherently allows bi-directional information flow, meaning that it can naturally merge information extracted from concrete data with highly-abstract prior knowledge. As an example of such kind of tasks, we report robust text recognition task with context information. We show that word spelling knowledge and word n-gram could be represented as a part of the network and actually they contribute the text recognition accuracy with noisy text images. We also show that the computational cost is approximately linear with the number of characters and words.

1 Introduction

Robust text recognition under highly noisy environment requires utilizing prior knowledge such as word spelling or word n-gram knowledge. One of the well know method is a technique that use weighted acyclic automata, also known as lattice, that represents various hypotheses form multiple stages [4]. The best path from top to bottom of the lattice represents the most probable candidate. To adopt this method for text recognition, first, we have to get candidates character set for each character image using some character recognition mechanism. And then we can construct a lattice using the candidate characters and prior knowledge. This is not ideal, because character recognition stage and text recognition stage are completely separated. This means that characters once omitted from the candidate set will never considered under the prior knowledge.

Hidemoto Nakada, Yuuji Ichisugi
National Institute of Advanced Science and Technology, 2-3-26 Aomi, Koto-ku, Tokyo, 135–0064 Japan, e-mail: {hide-nakada, y-ichisugi}@aist.go.jp

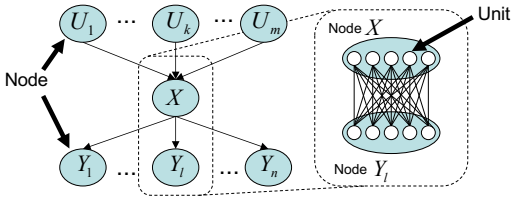

Fig. 1 BESOMNetwork. The ovals indicate the node and the small circles in oval stand for units. Typical BESOM network forms multi-layered structure as shown in this diagram.

We believe that Bayesian network based model is suitable for this kind of tasks, and have been developing BESOM model (BidirEctional Self Organizing Maps) [7][6]. BESOM is a machine learning model that models cerebral cortex as Baysian network. One of the characteristics of BESOM is the bi-directional information flow in the model, that enables hybrid-style inference using both of the concrete data and highly-abstracted prior knowledge[5]. This means that BESOM can interpret low-level sensor information using background knowledge which is extracted with other means.

In this paper, we discuss robust text recognition task using prior knowledge. We demonstrate that, 1) BESOM can easily represent prior knowledge such as word spell and word 2-gram as network structure, 2) BESOM can achieve better text-recognition accuracy, leveraging the prior knowledge, than state-of-the-art CNN, and 3) The computation cost is linear to the number of words and characters in the text.

2 Text recognition using context information with BESOM

2.1 BESOM

According to the recent studies in computational neuroscience, Bayesian network[8] can be the underlying mechanism of the cerebral cortex. Various neuroscientific phenomena are successfully reproduced with models based on Bayesian networks. The cerebral cortex shares many aspects, not only in functions but also in structures, with Bayesian networks. Based on this assumption, we have been developing BESOM, a Bayesian network based cerebral cortex model. BESOM represents macro columns in cerebral cortex as *nodes*, and mini columns as *units* in nodes. In Bayesian network terminology, nodes stands for probabilistic variables and units stands for specific values that the variable can take.

Naive Bayesian network is notorious for huge memory footprint and computation hungry nature. We have carefully restricted the conditional probability model to reduce the memory footprint and computational burden.

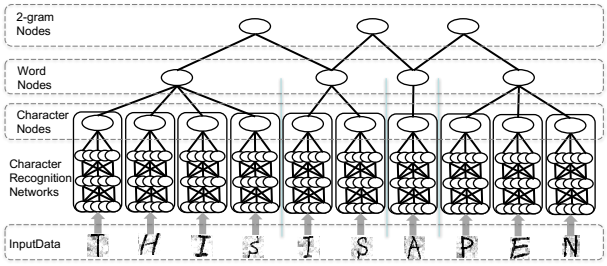

Fig. 2 Overview of the Network Structure.

2.2 Text Recognition with Prior Knowledge

Here, we discuss how to construct text recognition that leverages prior knowledge, especially word spell and word 2-gram knowledge. We assume that word spell and word 2-gram frequency is extracted from corpus in advance, and the character separation and word separation is performed in the previous step.

Figure 2 shows the overview of the whole network. We train character recognition network in advance and duplicate it n times, where n is the number of characters in the target text. On top of the character recognition networks, word spell network and word 2-gram network is attached. We call the top node of the character recognition network 'character node'.

A word spell node represents words with specific character length, meaning that node has number of words units, each of them corresponds to a specific word. Word spell node is connected to the character node, that has 27 units, each corresponds to a specific alphabet (one unit corresponds to 'cannot recognize'). Each units in word spell node and units in character node are connected with edges. The weight of the edge is 100% if the character corresponds to the character unit is included in the word corresponds to the word unit, otherwise 0%.

2-gram node represents two successive words with specific length combination and connected to the word nodes. Each unit represent specific successive words combination. All the units in 2-gram node are connected to all the units in the corresponding word nodes. The weights of the edge is 100% if the 2-gram uses the word, and 0% otherwise. Figure 3 shows the network. Note that only the 100% connections are shown.

To represent the occurrence frequency of 2-grams, we add one node per one 2-gram node (Figure 4). The node has just one unit that is always True which is connected to all the units in the 2-gram node by edges with weights equivalent to the occurrence frequency of the 2-gram.

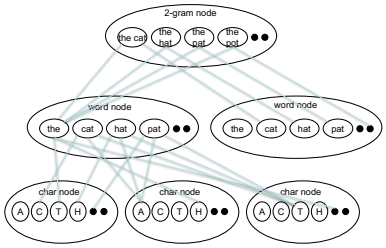

Fig. 3 Representation of Prior Knowledge.

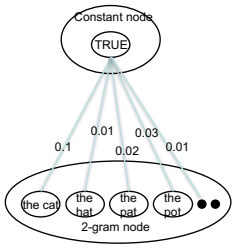

Fig. 4 Representation of 2-gram frequency.

3 Evaluation

We conducted experiments to confirm that prior knowledge could contributes robust text recognition under artificially added noise. We compared text recognition accuracy with three setups; namely, without prior knowledge, with word spell knowledge only, and with word spell knowledge and 2-gram.

The parameters of character recognition network in BESOM are as follows. The network has four layer in total, with the input layer, two hidden layers, and the output layer. Each node in the first hidden layer is connected to 4x4 region in the input layer and Each node in the second hidden layer is connected to 3x3 region in the first hidden layer, forming local receptive field. Number of nodes are 1024, 81, 9, and 1, and the number of units in a node are 2, 20, 100, and 27, respectively.

For comparison, we implemented conventional CNN (convolutional neural network) using Chainer[2], a neural network framework. The layer structure of the CNN is as follows; 32channels of 5x5 convolution, 2x2 max pooling, 64 channels of 5x5 convolution, 2x2 max pooling, 1024 channels of full connection layer, 26 channels of full connection layer. We employed ReLU as an activation function and dropout between the two full connection layers.

3.1 Evaluation Setup

We extracted words and word 2-grams from 3977 documents from RFC(Request For Comment)[3], the IETF standardization documents. We divide the documents into 80% of training data and 20% of test data. We extracted words and 2-grams from the training data. The number of 2-grams are 1,080,623.

We used ETL1[1] for the handwritten alphabet data, which has 32x32 pixels for each character. We just used 26 alphabet capitals and normalized them by cleaning noise and centering. Each alphabet had around 1,500 samples. We also divide them into training set and test set and train the BESOM character recognition network using the training set only.

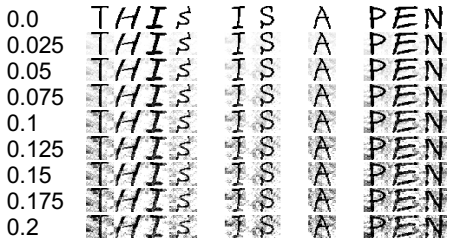

Fig. 5 Noise is generated following normal distribution with average 0.0 and distribution 1.0 multiplied by noise ratio. The noise is adopted repeatedly from each pixel and then 2x2 pixel areas, and then 4x4,.. and to 32x32 pixels. For the noise ratio, we tested from 0.0 to 0.2 by 0.025 step.

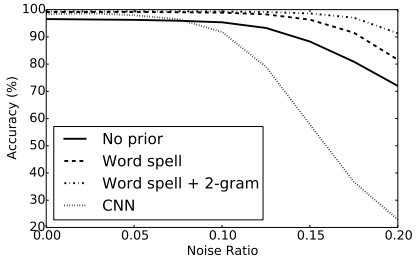

Fig. 6 Result of networks trained without noise.

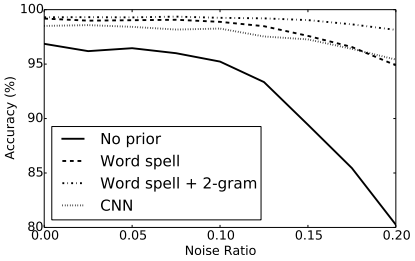

Fig. 7 Result of networks trained with noise.

For evaluation, we randomly pick 140 alphabets-only sentences from the text test set. The test text had 8133 characters in total. We rendered the evaluation texts using randomly picked handwritten alphabet from the test data set, adding noise as shown in Figure 5.

3.2 Evaluation Result

We performed two sets of evaluations. Using character recognition network trained using data with noise and without noise. Figure 6 shows the result using network trained noiseless data. The x-axis stands for the noise ratio and y-axis stands for the recognition accuracy. CNN shows drastic performance degradation along with increase of noise. The cause of this behavior considered to be the over-fitting to the noiseless data. In contrast, Bayesian network based models demonstrate stable performance even with the noisy data.

We can confirm that the word spell knowledge and word 2-gram knowledge drastically improve the recognition accuracy, showing 99% accuracy with noiseless data. With noise ratio 0.2, the model without prior knowledge shows 72% accuracy, while the model with spell knowledge shows 82% and with spell and 2-gram knowledge 91%.

Figure 7 shows the result for the experiment with models learned using data with noise. Note that the range of x-axis starts from 80%, showing significant im-

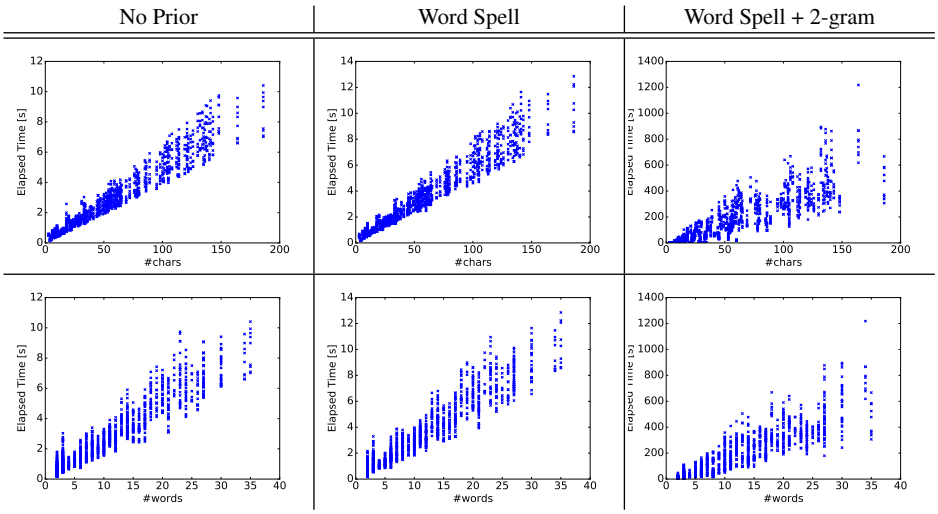

Fig. 8 Time to recognize text for number of characters (top raw) or words (bottom row) in the text. The left column shows the result without prior knowledge, the center column shows the result with word spell knowledge only, and the right column shows the result with word spell and word 2-gram knowledge.

provement in accuracy for all models. While the accuracy of CNN improved a lot, proposed method with word 2-gram knowledge outperformed it.

3.3 Time to Recognize Texts

Figure 8 shows the time to recognize text. Each dot represent one experiment with one text. The y-axis shows the time to execute the text recognition and the x-axis represents number of characters or number of words included in the target text. This evaluation is performed on a PC with two sockets of Intel(R) Xeon(R) CPU W5590 (3.33GHz, 4 core). However, the program uses just one core since the it is not parallelised yet.

We graphs show that the execution time is basically linear to the number of characters or words in the text. Adding to that, we can see that the word 2-gram is quite expensive in terms of computation. This is due to the fact that the large number of 2-grams means large number of units in the 2-gram nodes.

4 Conclusion

We have shown that robust text-recognition task could be solved with restricted Bayesian network that can represent prior knowledge in natural fashion. We have confirmed that, using prior knowledge, BESOM could outperform simple CNN in text recognition performance, and the computation cost is linear to the number of characters and words, although the cost is high.

Our future work includes speeding up the computation using grammatical knowledge on each word, which could dramatically reduce the amount of required computation.

Acknowledgements This paper is based on results obtained from a project commissioned by the New Energy and Industrial Technology Development Organization (NEDO).

References

1. About the etl character database : http://etlcdb.db.aist.go.jp/. Accessed: 2017-1-02
2. Chainer : http://chainer.org/. Accessed: 2017-1-02
3. Request for comments (rfc) : https://www.ietf.org/rfc.html. Accessed: 2017-1-02
4. Can, D., Narayanan, S.S.: A dynamic programming algorithm for computing n-gram posteriors from lattices. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 2388–2397. Association for Computational Linguistics (2015). URL aclweb.org/anthology/D/D15/D15-1286
5. George, D., Hawkins, J.: A hierarchical bayesian model of invariant pattern recognition in the visual cortex. In: Proc. of IJCNN 2005, vol. 3, p. 18121817 (2005)
6. Ichisugi, Y.: Regularization methods for the restricted bayesian network besom. In: Proc. of 23rd International Conference on Neural Information Processing (ICONIP2016). (2016)
7. Ichisugi, Y., Takahashi, N.: An efficient recognition algorithm for restricted bayesian networks. In: Proc. of 2015 International Joint Conference on Neural Networks (IJCNN 2015). (2015)
8. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann (1988)