# Summary of Character-level Convolutional Networks for Text Classification

Beom June Kim, Eunsun Lee

## 1 Introduction

CNN model is usually used for raw signals like images or speech recognition but in this paper, initially ConvNets is applied for character level text classification using one-dimensional ConvNets and large-scale datasets.
Applying ConvNets to text classification or large-scale natural language processing is shown not to require knowledge of the grammatical or meaning structure of the language.[dSG14][Kim14][JZ15] And Character-level ConvNets is not only inherently sequential, allowing them to be trained regardless of the possibility of word segmentation but also could train neologism like emoticons and spelling error.

## 2 Character-level Convolutional Networks

### 2.1 Key Modules

$$h(y) = \sum_{x=1}^{k} f(x) \cdot g(y \cdot d - x + c)$$

$$h(y) = \max_{x=1}^{k} g(y \cdot d - x + c)$$

$h_j(y)$ is a sum of the convolutions $g_i$(x) and $f_{ij}$(x) and implement max pooling. The model is applied the rectifier or thresholding function[NH10] for non-linearity and stochastic gradient descent(SGD)[Pol64][SMDH13].

### 2.2 Character quantization

Input as temporal data is encoded characters which is total 70 and one-hot encoding and the length is fixed $l_0$ involved blank. If input is over $l_0$, the excess is disregarded. Also Sequence for quantization is reversed so that it is straightforward for fully connected layers to combine weights with the most recent reading.

### 2.3 Model Design

This article performed a large and small version that are 6 convolutional layers and 3 fully connected layers. The number and length of input features are fixed at 70 and 1024. Also stride is 1 and dropout modules[HSK+12] are performed between fully connected layers to prevent over-fitting and back-propagation[Rum86] to optimize.
Word or sentence are replaced with synonyms using Thesaurus for augmentation that can be reduce generalization error.

## 3 Comparison Models

This article compare of another methods to present neutral information. the methods are Bag-of-words[Jon21] and its TFIDF, Bag-of-ngrams and its TFIDF, Bag-of-means word embedding as traditional methods. Word-based ConvNets[Kim14][MSC+13][CWB+11] and Long-Short Term Memory[GS05][GSK+15][HS97][
as deep learning methods.

# 4 Large-scale Datasets and Results

In this paper, the researchers have constructed various scales of datasets themselves. The types of datasets include news category classification, Chinese, Amazon review sentiment classification, and review ratings.

# 5 Discussion

| Model | AG | Sogou | DBP. | Yelp P. | Yelp F. | Yah. A. | Amz. F. | Amz. P. |
|---|---|---|---|---|---|---|---|---|
| BoW | 11.19 | 7.15 | 3.39 | 7.76 | 42.01 | 31.11 | 45.36 | 9.60 |
| BoW TFIDF | 10.36 | 6.55 | 2.63 | 6.34 | 40.14 | 28.96 | 44.74 | 9.00 |
| ngrams | 7.96 | 2.92 | 1.37 | **4.36** | 43.74 | 31.53 | 45.73 | 7.98 |
| ngrams TFIDF | **7.64** | **2.81** | **1.31** | 4.56 | 45.20 | 31.49 | 47.56 | 8.46 |
| Bag-of-means | **16.91** | **10.79** | **9.55** | **12.67** | **47.46** | **39.45** | **55.87** | **18.39** |
| LSTM | 13.94 | 4.82 | 1.45 | 5.26 | 41.83 | 29.16 | 40.57 | 6.10 |
| Lg. w2v Conv. | 9.92 | 4.39 | 1.42 | 4.60 | 40.16 | 31.97 | 44.40 | 5.88 |
| Sm. w2v Conv. | 11.35 | 4.54 | 1.71 | 5.56 | 42.13 | 31.50 | 42.59 | 6.00 |
| Lg. w2v Conv. Th. | 9.91 | - | 1.37 | 4.63 | 39.58 | 31.23 | 43.75 | 5.80 |
| Sm. w2v Conv. Th. | 10.88 | - | 1.53 | 5.36 | 41.09 | 29.86 | 42.50 | 5.63 |
| Lg. Lk. Conv. | 8.55 | 4.95 | 1.72 | 4.89 | 40.52 | 29.06 | 45.95 | 5.84 |
| Sm. Lk. Conv. | 10.87 | 4.93 | 1.85 | 5.54 | 41.41 | 30.02 | 43.66 | 5.85 |
| Lg. Lk. Conv. Th. | 8.93 | - | 1.58 | 5.03 | 40.52 | 28.84 | 42.39 | 5.52 |
| Sm. Lk. Conv. Th. | 9.12 | - | 1.77 | 5.37 | 41.17 | 28.92 | 43.19 | 5.51 |
| Lg. Full Conv. | 9.85 | 8.80 | 1.66 | 5.25 | 38.40 | 29.90 | 40.89 | 5.78 |
| Sm. Full Conv. | 11.59 | 8.95 | 1.89 | 5.67 | 38.82 | 30.01 | 40.88 | 5.78 |
| Lg. Full Conv. Th. | 9.51 | - | 1.55 | 4.88 | 38.04 | 29.58 | 40.54 | 5.51 |
| Sm. Full Conv. Th. | 10.89 | - | 1.69 | 5.42 | **37.95** | 29.90 | 40.53 | 5.66 |
| Lg. Conv. | 12.82 | 4.88 | 1.73 | 5.89 | 39.62 | 29.55 | 41.31 | 5.51 |
| Sm. Conv. | 15.65 | 8.65 | 1.98 | 6.53 | 40.84 | 29.84 | 40.53 | 5.50 |
| Lg. Conv. Th. | 13.39 | - | 1.60 | 5.82 | 39.30 | **28.80** | 40.45 | **4.93** |
| Sm. Conv. Th. | 14.80 | - | 1.85 | 6.49 | 40.16 | 29.84 | **40.43** | 5.67 |

Figure 1: Testing errors of all the models

As a result of training all the models in the experiment, the outcomes were derived as shown in Figure 1. The first row of the chart represents the datasets constructed by the researchers, with the scale of the datasets increasing as one moves to the right. The first column of the chart represents the model names, where 'Lg' stands for Large, 'Sm' for Small, 'LK' for lookup, 'w2v' for Word2Vector, and 'Th' for Thesaurus. If 'Full' is present, it indicates that case sensitivity was maintained. Based on the results of the chart above, the following conclusions can be drawn.

1. Character-level ConvNet is an less resource-intensive model.
2. The ConvNet model showed better performance than traditional models when the dataset was larger.
3. The ConvNet model showed better performance with datasets created by actual users.
4. Whether or not to distinguish between uppercase and lowercase letters showed different results depending on the model, but not distinguishing them showed better performance in large-scale datasets.
5. The dichotomous classification was not important in choosing which model to use.
6. The Bag-of-Means method should be avoided in word2vec.
7. To find a model that performs well, all models must be tested.

# 6 Conclusions and Outlook

This paper compares a large number of traditional models and deep learning models using various large-scale datasets, demonstrating that Character-level ConvNet is an effective method. How well this model performs is determined by various factors such as the size of the dataset, whether text preprocessing is done, and the choice of alphabet.

# References


[CWB+11] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel P. Kuksa. Natural language processing (almost) from scratch. *CoRR*, abs/1103.0398, 2011.

[dSG14] Cícero dos Santos and Maíra Gatti. Deep convolutional neural networks for sentiment analysis of short texts. pages 69–78, August 2014.

[GS05] Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5):602–610, 2005. IJCNN 2005.

[GSK+15] Klaus Greff, Rupesh Kumar Srivastava, Jan Koutník, Bas R. Steunebrink, and Jürgen Schmidhuber. LSTM: A search space odyssey. *CoRR*, abs/1503.04069, 2015.

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 11 1997.

[HSK+12] Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *CoRR*, abs/1207.0580, 2012.

[Jon21] Karen Spärck Jones. A statistical interpretation of term specificity and its application in retrieval. *J. Documentation*, 60:493–502, 2021.

[JZ15] Rie Johnson and Tong Zhang. Effective use of word order for text categorization with convolutional neural networks, 2015.

[Kim14] Yoon Kim. Convolutional neural networks for sentence classification. pages 1746–1751, October 2014.

[MSC+13] Tomás Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. *CoRR*, abs/1310.4546, 2013.

[NH10] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, page 807–814, Madison, WI, USA, 2010. Omnipress.

[PMB12] Razvan Pascanu, Tomás Mikolov, and Yoshua Bengio. Understanding the exploding gradient problem. *CoRR*, abs/1211.5063, 2012.

[Pol64] B.T. Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964.

[Rum86] Hinton G. Williams R. Rumelhart, D. Learning representations by back-propagating errors. *Nature 323*, page 533–536, 1986.

[SMDH13] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, ICML'13, page III–1139–III–1147. JMLR.org, 2013.