

End-to-End Multi-View Networks for Text Classification

Hongyu Guo and Colin Cherry and Jiang Su

National Research Council Canada

1200 Montreal Road, Ottawa, Ontario, K1A 0R6, Canada

firstname.lastname@nrc-cnrc.gc.ca

Abstract

We propose a multi-view network for text classification. Our method automatically creates various views of its input text, each taking the form of soft attention weights that distribute the classifier’s focus among a set of base features. For a bag-of-words representation, each view focuses on a different subset of the text’s words. Aggregating many such views results in a more discriminative and robust representation. Through a novel architecture that both stacks and concatenates views, we produce a network that emphasizes both depth and width, allowing training to converge quickly. Using our multi-view architecture, we establish new state-of-the-art accuracies on two benchmark tasks.

1 Introduction

State-of-the-art deep neural networks leverage task-specific architectures to develop hierarchical representations of their input, with each layer building a refined abstraction of the layer that came before it (Conneau et al., 2016). For text classification, one can think of this as a single reader building up an increasingly refined understanding of the content. In a departure from this philosophy, we propose a divide-and-conquer approach, where a team of readers each focus on different aspects of the text, and then combine their representations to make a joint decision.

More precisely, the proposed Multi-View Network (MVN) for text classification learns to generate several views of its input text. Each view is formed by focusing on different sets of words through a view-specific attention mechanism. These views are arranged sequentially, so each subsequent view can build upon or deviate

from previous views as appropriate. The final representation that concatenates these diverse views should be more robust to noise than any one of its components. Furthermore, different sentences may look similar under one view but different under another, allowing the network to devote particular views to distinguishing between subtle differences in sentences, resulting in more discriminative representations.

Unlike existing multi-view neural network approaches for image processing (Zhu et al., 2014; Su et al., 2015), where multiple views are provided as part of the input, our MVN learns to automatically create views from its input text by focusing on different sets of words. Compared to deep Convolutional Networks (CNN) for text (Zhang et al., 2015; Conneau et al., 2016), the MVN strategy emphasizes network width over depth. Shorter connections between each view and the loss function enable better gradient flow in the networks, which makes the system easier to train. Our use of multiple views is similar in spirit to the weak learners used in ensemble methods (Breiman, 1996; Friedman et al., 1998; Wolpert, 1992), but our views produce vector-valued intermediate representations instead of classification scores, and all our views are trained jointly with feedback from the final classifier.

Experiments on two benchmark data sets, the Stanford Sentiment Treebank (Socher et al., 2013) and the AG English news corpus (Zhang et al., 2015), show that 1) our method achieves very competitive accuracy, 2) some views distinguish themselves from others by better categorizing specific classes, and 3) when our base bag-of-words feature set is augmented with convolutional features, the method establishes a new state-of-the-art for both data sets.

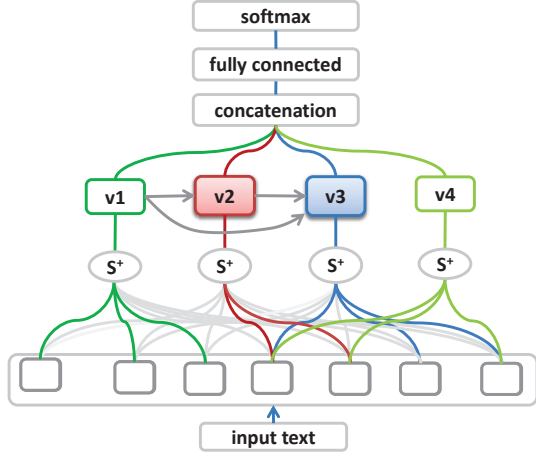


Figure 1: A MVN architecture with four views.

2 Multi-View Networks for Text

The MVN architecture is depicted in Figure 1. First, individual *selection* vectors s^+ are created, each formed by a distinct softmax weighted sum over the word vectors of the input text. Next, these selections are sequentially transformed into *views* v , with each view influencing the views that come after it. Finally, all views are concatenated and fed into a two-layer perceptron for classification.

2.1 Multiple Attentions for Selection

Each selection s^+ is constructed by focusing on a different subset of words from the original text, as determined by a softmax weighted sum (Bahdanau et al., 2014). Given a piece of text with H words, we represent it as a bag-of-words feature matrix $B \in \mathbb{R}^{H \times d}$. Each row of the matrix corresponds to one word, which is represented by a d -dimensional vector, as provided by a learned word embedding table. The selection s_i^+ for the i^{th} view is the softmax weighted sum of features:

$$s_i^+ = \sum_{h=1}^H d_{i,h} B[h : h] \quad (1)$$

where the weight $d_{i,h}$ is computed by:

$$d_{i,h} = \frac{\exp(m_{i,h})}{\sum_{h=1}^H \exp(m_{i,h})} \quad (2)$$

$$m_{i,h} = w_i^s \tanh(W_i^s B[h : h]) \quad (3)$$

here, w_i^s (a vector) and W_i^s (a matrix) are learned selection parameters. By varying the weights $d_{i,h}$, the selection for each view can focus on different words from B , as illustrated by different color curves connecting to s^+ in Figure 1.

2.2 Aggregating Selections into Views

Having built one s^+ for each of our V views, the actual views are then created as follows:

$$v_1 = s_1^+; v_V = s_V^+ \quad (4)$$

$$v_i = \tanh(W_i^v([v_1; v_2; \dots; v_{i-1}; s_i^+])) \quad (5)$$

for $i = 2 \dots V - 1$

where W_i^v are learned parameter matrices, and $[\dots; \dots]$ represents concatenation. The first and last views are formed by solely s^+ ; however, they play very different roles in our network. v_V is completely disconnected from the others, an independent attempt at good feature selection, intended to increase view diversity (Muslea et al., 2002; Guo and Viktor, 2006, 2008; Wang et al., 2015). Conversely, v_1 forms the base of a structure similar to a multi-layer perceptron with short-cutting, as defined by the recurrence in Equation 5. Here, the concatenation of all previous views implements short-cutting, while the recursive definition of each view implements stacking, forming a deep network depicted by horizontal arrows in Figure 1. This structure makes each view aware of the information in those previous to it, allowing them to build upon each other. Note that the W^v matrices are view-specific and grow with each view, making the overall parameter count quadratic in the number of views.

2.3 Classification with Views

The final step is to transform our views into a classification of the input text. The MVN does so by concatenating its view vectors, which are then fed into a fully connected projection followed by a softmax function to produce a distribution over the possible classes. Dropout regularization (Hinton et al., 2012) can be applied at this softmax layer, as in (Kim, 2014).

2.4 Beyond Bags of Words

The MVN's selection layer operates on a matrix of feature vectors B , which has thus far corresponded to a bag of word vectors. Each view's selection makes intuitive sense when features correspond to words, as it is easy to imagine different readers of a text focusing on different words, with each reader arriving at a useful interpretation. However, there is a wealth of knowledge on how to construct powerful feature representations for text, such as those used by convolutional neural networks (CNNs). To demonstrate the utility of

having views that weight arbitrary feature vectors, we augment our bag-of-words representation with vectors built by n -gram filters max-pooled over the entire text (Kim, 2014), with one feature vector for each n -gram order, $n = 2 \dots 5$. The augmented B matrix has $H + 4$ rows. Unlike our word vectors, the 4 CNN vectors each provide representations of the entire text. Returning to our reader analogy, one could imagine these to correspond to quick ($n = 2$) or careful ($n = 5$) skims of the text. Regardless of whether a feature vector is built by embedding table or by max-pooled n -gram filters, we always back-propagate through all feature construction layers, so they become specialized to our end task.

3 Experiments

3.1 Stanford Sentiment Treebank

The Stanford Sentiment Treebank contains 11,855 sentences from movie reviews. We use the same splits for training, dev, and test data as in (Socher et al., 2013) to predict the fine-grained 5-class sentiment categories of the sentences. For comparison purposes, following (Kim, 2014; Kalchbrenner et al., 2014; Lei et al., 2015), we train the models using both phrases and sentences, but only evaluate sentences at test time.

We initialized all of the word embeddings (Cherry and Guo, 2015; Cherry et al., 2015) using the publicly available 300 dimensional pre-trained vectors from GloVe (Pennington et al., 2014). We learned 8 views with 200 dimensions each, which requires us to project the 300 dimensional word vectors, which we implemented using a linear transformation, whose weight matrix and bias term are shared across all words, followed by a tanh activation. For optimization, we used Adadelta (Zeiler, 2012), with a starting learning rate of 0.0005 and a mini-batch of size 50. Also, we used dropout (with a rate of 0.2) to avoid overfitting. All of these MVN hyperparameters were determined through experiments measuring validation-set accuracy.

The test-set accuracies obtained by different learning methods, including the current state-of-the-art results, are presented in Table 1. The results indicate that the bag-of-words MVN outperforms most methods, but obtains lower accuracy than the state-of-the-art results achieved by the tree-LSTM (Tai et al., 2015; Zhu et al., 2015) and the high-order CNN (Lei et al., 2015). However,

MVN (with convolutional features)	51.5
MVN	49.6
high-order CNN	51.2
tree-LSTM	51.0
DRNN	49.8
DCNN	48.5
CNN-MC	47.4
NBoW	44.5
SVM	38.3

Table 1: Accuracies on the Stanford Sentiment Treebank 5-class classification task; except for the MVN, all results are drawn from (Lei et al., 2015).

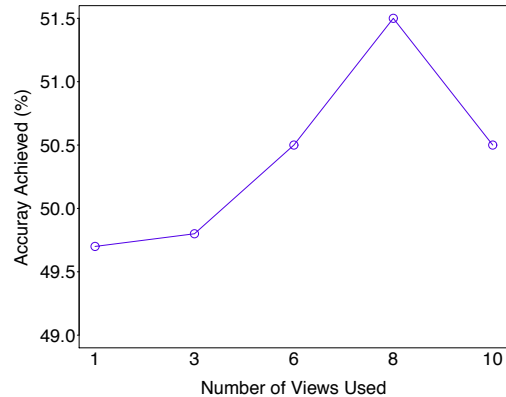


Figure 2: Accuracies obtained by varying the number of views.

when augmented with 4 convolutional features as described in Section 2.4, the MVN strategy surpasses both of these, establishing a new state-of-the-art on this benchmark.

In Figure 2, we present the test-set accuracies obtained while varying the number of views in our MVN with convolutional features. These results indicate that better predictive accuracy can be achieved while increasing the number of views up to eight. After eight, the accuracy starts to drop. The number of MVN views should be tuned for each new application, but it is good to see that not too many views are required to achieve optimal performance on this task.

To better understand the benefits of the MVN method, we further analyzed the eight views constructed by our best model. After training, we obtained the view representation vectors for both the training and testing data, and then independently trained a very simple, but fast and stable Naïve Bayes classifier (McCallum and Nigam, 1998) for each view. We report class-specific F-measures for

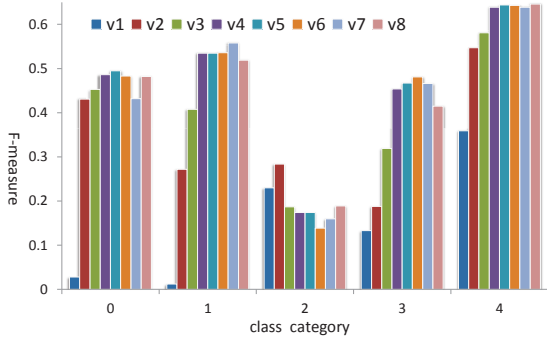


Figure 3: Class-specific F-measures obtained by Naïve Bayes classifiers built over different views.

Full MVN	51.5
Voting by 8 independent, 1-view MVNs (weak-learner: 49.5 ± 0.20)	50.2
MVN w/ no horizontal links	49.0
MVN w/ length-1 horizontal links	50.5

Table 2: Ablation experiments on the Stanford Sentiment Treebank test set

each view in Figure 3. From this figure, we can observe that different views focus on different target classes. For example, the first two views perform poorly on the 0 (very negative) and 1 (negative) classes, but achieve the highest F-measures on the 2 (neutral) class. Meanwhile, the non-neutral classes each have a different view that achieves the highest F-measure. This suggests that some views have specialized in order to better separate subsets of the training data.

We provide an ablation study in Table 2. First, we construct a traditional ensemble model. We independently train eight MVN models, each with a single view, to serve as weak learners. We have them vote with equal weight for the final classification, obtaining a test-set accuracy of 50.2. Next, we restrict the views in the MVN to be unaware of each other. That is, we replace Equation 5 with $v_i = s_i^+$, which removes all horizontal links in Figure 1. This drops performance to 49.0. Finally, we experiment with a variant of MVN, where each view is only connected to the most recent previous view, replacing Equation 5 with $v_i = \tanh(W_i^v([v_{i-1}; s_i^+]))$, leading to a version where the parameter count grows linearly in the number of views. This drops the test-set performance to 50.5. These experiments suggest that enabling the views to build upon each other is crucial for achieving the best performance.

MVN (with convolutional features)	7.13
MVN	7.49
n -grams TFIDF	7.64
n -grams	7.96
Lg. Lk. Convolution	8.55
29 layers Convolution with KMaxPooling	8.67
Lg. Full Convolution	9.85
BoW	11.19
LSTM	13.94
Bag-of-means	16.91

Table 3: Error rates on the AG News test set. All results except for the MVN are drawn from (Conneau et al., 2016)

3.2 AG’s English News Categorization

The AG corpus (Zhang et al., 2015; Conneau et al., 2016) contains categorized news articles from more than 2,000 news outlets on the web. The task has four classes, and for each class there are 30,000 training documents and 1,900 test documents. A random sample of the training set was used for hyper-parameter tuning. The training and testing settings of this task are exactly the same as those presented for the Stanford Sentiment Treebank task in Section 3.1, except that the mini-batch size is reduced to 23, and each view has a dimension of 100.

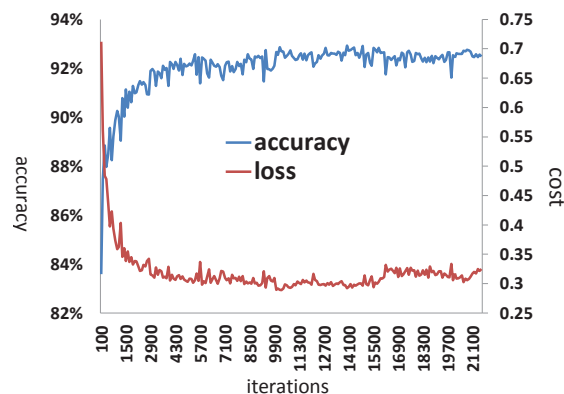
The test errors obtained by various methods are presented in Table 3. These results show that the bag-of-words MVN outperforms the state-of-the-art accuracy obtained by the non-neural n -gram TFIDF approach (Zhang et al., 2015), as well as several very deep CNNs (Conneau et al., 2016). Accuracy was further improved when the MVN was augmented with 4 convolutional features.

In Figure 4, we show how accuracy and loss evolve on the validation set during MVN training. These curves show that training is quite stable. The MVN achieves its best results in just a few thousand iterations.

4 Conclusion and Future Work

We have presented a novel multi-view neural network for text classification, which creates multiple views of the input text, each represented as a weighted sum of a base set of feature vectors. These views work together to produce a discriminative feature representation for text classification. Unlike many neural approaches to classification,

Figure 4: Accuracies and cost on the validation set during training on the AG News data set.



our architecture emphasizes network width in addition to depth, enhancing gradient flow during training. We have used the multi-view network architecture to establish new state-of-the-art results on two benchmark text classification tasks. In the future, we wish to better understand the benefits of generating multiple views, explore new sources of base features, and apply this technique to other NLP problems such as translation or tagging.

References

- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR* abs/1409.0473.
- Leo Breiman. 1996. Bagging predictors. *Mach. Learn.* 24(2).
- Colin Cherry and Hongyu Guo. 2015. The unreasonable effectiveness of word representations for twitter named entity recognition. In *HLT-NAACL*. pages 735–745.
- Colin Cherry, Hongyu Guo, and Chengbi Dai. 2015. Nrc: Infused phrase vectors for named entity recognition in twitter. *ACL-IJCNLP 2015*:54–60.
- Alexis Conneau, Holger Schwenk, Loïc Barrault, and Yann LeCun. 2016. Very deep convolutional networks for natural language processing. *CoRR* abs/1606.01781.
- Jerome Friedman, Trevor Hastie, and Robert Tibshirani. 1998. Additive logistic regression: a statistical view of boosting. *Annals of Statistics* 28:2000.
- Hongyu Guo and Herna L Viktor. 2006. Mining relational data through correlation-based multiple view validation. In *KDD’06*. ACM, pages 567–573.
- Hongyu Guo and Herna L Viktor. 2008. Multirelational classification: a multiple view approach. *Knowledge and Information Systems* 17(3):287–312.
- Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *CoRR* abs/1207.0580.
- Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. *CoRR* abs/1404.2188.
- Yoon Kim. 2014. Convolutional neural networks for sentence classification. *CoRR* abs/1408.5882.
- Tao Lei, Regina Barzilay, and Tommi S. Jaakkola. 2015. Molding cnns for text: non-linear, non-consecutive convolutions. *CoRR* abs/1508.04112.
- Andrew McCallum and Kamal Nigam. 1998. A comparison of event models for naive bayes text classification. *AAAI Workshop*.
- Ion Muslea, Steven Minton, and Craig A. Knoblock. 2002. Active + semi-supervised learning = robust multi-view learning. In *Proceedings of the Nineteenth International Conference on Machine Learning*. ICML ’02, pages 435–442.
- Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*. volume 14, pages 1532–43.
- Richard Socher, Alex Perelygin, Jean Y. Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP ’13*.
- Hang Su, Subhansu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. 2015. Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*. ICCV ’15, pages 945–953.
- Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. *CoRR* abs/1503.00075.
- Weiran Wang, Raman Arora, Karen Livescu, and Jeff Bilmes. 2015. On deep multi-view representation learning. In *International Conference on Machine Learning (ICML)*. Lille, France.
- David H. Wolpert. 1992. Stacked generalization. *Neural Networks* 5:241–259.
- Matthew D. Zeiler. 2012. ADADELTA: an adaptive learning rate method. *CoRR* abs/1212.5701.

- Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *NIPS'15*. pages 649–657.
- Xiaodan Zhu, Parinaz Sobhani, and Hongyu Guo. 2015. Long short-term memory over recursive structures. In *ICML*. pages 1604–1612.
- Zhenyao Zhu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. 2014. Multi-view perceptron: a deep model for learning face identity and view representations. In *NIPS*. pages 217–225.