Variational Self-attention Model for Sentence Representation

Qiang Zhang[1], Shangsong Liang[2], Emine Yilmaz[1]
[1] University College London, London, United Kingdom
[2] Sun Yat-sen University, Guangzhou, China
{qiang.zhang.16, emine.yilmaz}@ucl.ac.uk, liangshangsong@gmail.com

Abstract

This paper proposes a variational self-attention model (VSAM) that employs variational inference to derive self-attention. We model the self-attention vector as random variables by imposing a probabilistic distribution. The self-attention mechanism summarizes source information as an attention vector by a weighted sum, where the weights are a learned probabilistic distribution. Compared with conventional deterministic counterpart, the stochastic units incorporated by VSAM allow multi-modal attention distributions. Furthermore, by marginalizing over the latent variables, VSAM is more robust against overfitting. Experiments on the stance detection task demonstrate the superiority of our method.

1 Background

1.1 Sentence representation

A sentence usually consists of a sequence of discrete words or tokens $v = [v_1, v_2, \dots, v_n]$, where v_i can be a one-hot vector with the dimension N equal to the number of unique tokens in the vocabulary. Pre-trained distributed word embeddings, such as Word2vec [6] and GloVe [8], have been developed to transform v_i into a lower-dimensional vector representation x_i, whose dimension D is much smaller than N. Thus, a sentence can be encoded in a more dense representation $x = [x_1, x_2, \dots, x_n]$. The encoding process can be written as: $x = W^e v$, where W^e is the transformation matrix. In the areas of natural language processing, the majority of deep learning methods (e.g. RNN and CNN) take x as the input and generate a compact vector representation s for a sentence: $s = f^{\text{RNN}}(x)$, where $f^{\text{RNN}}(\cdot)$ indicates a RNN model. These methods consider the semantic dependencies between x_i and its context and hence believe that s summarizes the semantic information of the entire sentence.

1.2 Self-attention

The attention mechanism [1, 9] has been proposed as an alignment score between elements from two vector representations. Specifically, given the vector representation of a query q and a token sequence $x = [x_1, x_2, \dots, x_n]$, the attention mechanism is to compute the alignment score between x_i and q.

Self-attention [7] is a special case of the attention mechanism, where q is replaced with a token embedding x_j from the input sequence itself. Self-attention is a method of encoding sequences of vectors by relating these vectors to each-other based on pairwise similarities. It measures the dependency between each pair of tokens, x_i and x_j, from the same input sequence: $a_{i,j} = f^{\text{self-attention}}(x_i, x_j)$, where $f^{\text{self-attention}}(\cdot, \cdot)$ indicates a self-attention implementation.

Self-attention is very expressive and flexible for both long-term and local dependencies, which used to be respectively modeled by RNN and CNN. Moreover, the self-attention mechanism has fewer parameters and faster convergence than RNN. Recently, a variety of NLP tasks have experienced improvement brought by the self-attention mechanism.

Third workshop on Bayesian Deep Learning (NeurIPS 2018), Montréal, Canada.

1.3 Neural variational inference

Latent variable modeling is popular for many NLP tasks [5, 2]. It populates hidden representations to a region (in stead of a single point), making it possible to generate diversified data from the vector space or even control the generated samples. It is non-trivial to carry out effective and efficient inference for complex and deep models. Training neural networks as powerful function approximators through backpropagation has given rise to promising frameworks to latent variable modeling [3, 4].

The modeling process builds a generative model and an inference model. A generative model is to construct the joint distribution and somehow capture the dependencies between variables. For a generative model with a latent variable z, it can be seen as stochastic units in deep neural networks. We define the observed parent and child nodes of z as x and y respectively. Hence the joint distribution of the generative model is:

$$p_\theta(x,y) = \int p_\theta(y|z)p_\theta(z|x)p(x)dz, \tag{1}$$

where θ parameters the generative distributions $p_\theta(y|z)$ and $p_\theta(z|x)$. The variational lower bound is:

$$\begin{aligned}\mathscr{L} &= \mathbb{E}_{q_\phi(z)}[\log(p_\theta(y|z)p_\theta(z|x)p(x)) - \log q_\phi(z)] = \int \log \frac{p_\theta(y|z)p_\theta(z|x)p(x)}{q_\phi(z)} q_\phi(z)dz \\ &\leqslant \log \int p_\theta(y|z)p_\theta(z|x)p(x)dz = \log p_\theta(x,y) \end{aligned} \tag{2}$$

In order to derive a tight lower bound, the variational distribution $q_\phi(z)$ should approach the true posterior distribution $p_\theta(z|x,y)$. A parametrized diagonal Gaussian distribution $\mathcal{N}(z|\mu(x,y), \text{diag}(\sigma^2(x,y)))$ is employed as $q_\phi(z|x,y)$.

The inference model is to derive the variational distribution that approaches the posterior distribution of latent variables given observed variables. The three steps to construct the inference model are:

1. Construct vector representations of the observed variables: $u = f_x(x)$, $v = f_y(y)$.
2. Assemble a joint distribution: $\pi = g(u,v)$.
3. Parameterize the variational distribution over the latent variables: $\mu = l_1(\pi)$, $\log\sigma = l_2(\pi)$.

$f^x(\cdot)$ and $f^y(\cdot)$ can be any type of deep neural networks that are suitable for the observed data; $g(\cdot)$ is an MLP that concatenates the vector representations of the conditioning variables; $l(\cdot)$ is a linear transformation which outputs the parameters of the Gaussian distribution. By sampling from the variational distribution, $z \sim q_\phi(z|x,y)$, we are able to carry out stochastic back-propagation to optimize the lower bound.

During the training process, the generative model parameters θ together with the inference model parameters ϕ are updated by stochastic back-propagation based on samples z drawn from $q_\phi(z|x,y)$. Let L denote the total number of samples. For the gradients w.r.t. θ, we have the form:

$$\nabla_\theta \mathscr{L} \simeq \frac{1}{L}\sum_{l=1}^{L} \nabla_\theta \log(p_\theta(y|z^{(l)})p_\theta(z^{(l)}|x)) \tag{3}$$

For the gradients w.r.t. parameters ϕ, we reparameterize $z^{(l)} = \mu + \sigma \cdot \epsilon^{(l)}$ and samples $\epsilon^{(l)} \sim \mathcal{N}(0, I)$ to reduce the variance in stochastic estimation. The update of ϕ can be carried out by back-propagating the gradients w.r.t. μ and σ:

$$\gamma(z) = \log(p_\theta(y|z)p_\theta(z|x)) - \log q_\phi(z|x,y) \tag{4}$$

$$\nabla_\mu \mathscr{L} \simeq \frac{1}{L}\sum_{l=1}^{L} \nabla_{z^{(l)}}[\gamma(z^{(l)})], \nabla_\sigma \mathscr{L} \simeq \frac{1}{2L}\sum_{l=1}^{L} \epsilon^{(l)} \nabla_{z^{(l)}}[\gamma(z^{(l)})] \tag{5}$$

It is worth mentioning that unsupervised learning is a special case of the neural variational framework where z has no parent node x. In that case z is directly drawn from the prior $p(z)$ instead of the conditional distribution $p(z|x)$, and $\gamma(z) = \log(p_\theta(y|z)p_\theta(z)) - \log q_\phi(z|y)$.

Here we only discuss the scenario where the latent variables are continuous and the parameterized diagonal Gaussian is employed as the variational distribution. However the framework is also suitable for discrete units, and the only modification needed is to replace the Gaussian with a multinomial parameterized by the outputs of a softmax function. Though the reparameterization trick for continuous variables is not applicable for this case, a policy gradient approach (Mnih & Gregor, 2014) can help to alleviate the high variance problem during stochastic estimation.

2 Variational Self-attention Model

In this paper we propose a Variational Self-attention Model (VSAM) that employs variational inference to learn self-attention. In doing so the model will implement a stochastic self-attention learning mechanism instead of the conventional deterministic one, and obtain a more salient inner-sentence semantic relationship. The framework of the model is shown in Figure 1.

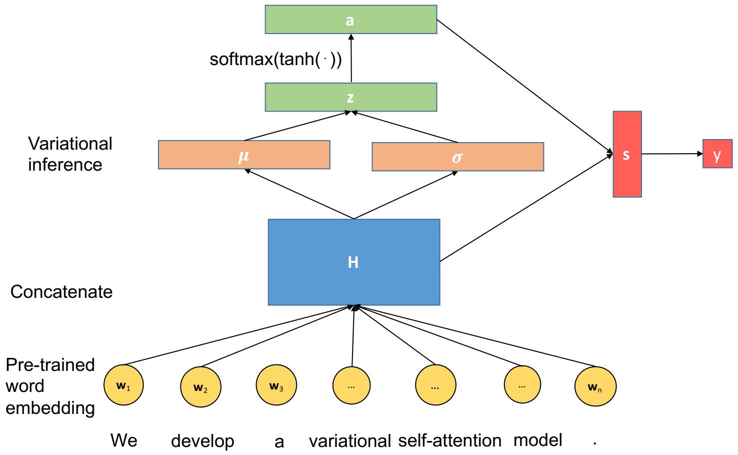

Figure 1: The general framework of the variational self-attention model for sentence representation.

Suppose we have a sentence $x = [x_1, x_2, \ldots, x_n]$, where x_i is the pre-trained word embedding and n is the number of words in the sentence. We concatenate the word embeddings to form a matrix $H \in \mathbb{R}^{D\times n}$, where D is the dimension of the word embedding. We aim to learn semantic dependencies between every pair of tokens through self-attention. Instead of using the deterministic self-attention vector, VSAM employs a latent distribution $p_\theta(z|H)$ to model semantic dependencies, which is a parameterized diagonal Gaussian $\mathcal{N}(z|\mu(H), \mathrm{diag}(\sigma^2(H)))$. Therefore, the self-attention model extracts an attention vector a based on the stochastic vector $z \sim p_\theta(z|H)$.

The diagonal Gaussian conditional distribution $p_\theta(z|H)$ can be calculated as follows:

$$\pi_\theta = f_\theta(H) \tag{6}$$

$$\mu_\theta = l_1(\pi_\theta), \log\sigma_\theta = l_2(\pi_\theta) \tag{7}$$

$$p_\theta(z|H) = \mathcal{N}(\mu_\theta, \mathrm{diag}(\sigma_\theta^2)). \tag{8}$$

For each sentence embedding H, the neural network generates the corresponding parameters μ_θ and σ_θ that parametrize the latent self-attention distribution over the entire sentence semantics.

The self-attention vector $a \in \mathbb{R}^{n\times 1}$ can then be derived as: $a = \mathrm{softmax}(\tanh(W^z z))$. The final sentence vector representation s is the sentence embedding matrix H weighted by the self-attention vector a as: $s = Ha$, where $s \in \mathbb{R}^{D\times 1}$. For the downstream application with expected output y, the conditional probability distribution $p_\theta(y|s)$ can be modeled as: $p_\theta(y|s) = g_\theta(s)$. As for the inference network, we follow the neural variational inference framework and construct a deep neural network as the inference network. We use H and y to compute $q_\phi(z|H, y)$ as: $\pi_\phi = f_\phi(H, y)$. According to the joint representation π_ϕ, we can then generate the parameters μ_ϕ and σ_ϕ, which parameterize the variational distribution over the sentence semantics z:

$$\mu_\phi = l_3(\pi_\phi), \log\sigma_\phi = l_4(\pi_\phi) \tag{9}$$

$$q_\phi(z|H, y) = \mathcal{N}(\mu_\phi, \mathrm{diag}(\sigma_\phi^2)). \tag{10}$$

Table 1: Statistics of the FNC-1 dataset.

Stance	Training		Test	
	Number	Percentage	Number	Percentage
agree	3,678	7.36	1,903	7.49
disagree	840	1.68	697	2.74
discuss	8,909	17.83	4,464	17.57
unrelated	36,545	73.13	18,349	72.20
	49,972		25,413	

Table 2: Performance comparison with the state-of-art algorithms on the FNC-1 test dataset.

Model	Accuracy (%)				Micro F1(%)
	agree	disagree	discuss	unrelated	
Average of Word2vec Embedding	12.43	1.30	43.32	74.24	45.53
CNN-based Sentence Embedding	24.54	5.06	53.24	79.53	81.72
RNN-based Sentence Embedding	24.42	5.42	69.05	65.34	78.70
Self-attention Sentence Embedding	23.53	4.63	63.59	80.34	80.11
Our model	28.53	10.43	65.43	82.43	**83.54**

To emphasize, although both $p_\theta(z|H)$ and $q_\phi(z|H, y)$ are modeled as parameterized Gaussian distributions, $q_\phi(z|H, y)$ as an approximation only functions during inference by producing samples to compute the stochastic gradients, while $p_\theta(z|H)$ is the generative distribution that generates the samples for predicting y. To maximize the log-likelihood $\log p(y|H)$ we use the variational lower bound. Based on the samples $z \sim q_\phi(z|H, y)$, the variational lower bound can be derived as

$$\begin{aligned} \mathscr{L} &= \mathbb{E}_{q_\phi(z|H,y)}[\log p_\theta(y|H)] - D_{\mathrm{KL}}(q_\phi(z|H,y)||p_\theta(z|H)) \\ &\leqslant \log \int p_\theta(y|z)p_\theta(z|H)dz = \log p(y|H). \end{aligned} \tag{11}$$

The generative model parameters θ and the inference model parameters ϕ are updated jointly according to their stochastic gradients. In this case, $D_{\mathrm{KL}}(q_\phi(z|H, y)||p_\theta(z|H))$ can be analytically computed during the training process.

3 Experiments

In this section, we describe our experimental setup. The task we address is to detect the stance of a piece of text towards a claim as one of the four classes: *agree*, *disagree*, *discuss* and *unrelated* [10]. Experiments are conducted on the FNC-1 official dataset [1]. The dataset are split into training and testing subsets, respectively; see Table 1 for statistics of the split. We report classification accuracy and micro F1 metrics on test dataset for each type of stances.

Baselines for comparisons include: (1) **Average of Word2vec Embedding** refers to sentence embedding by averaging vectors of each word based on Word2vec. (2) **CNN-based Sentence Embedding** refers to sentence embedding by inputting the Word2vec embedding of each word to a convolutional neural network. (3) **Self-attention Sentence Embedding** refers to sentence embedding by calculating self-attention based sentence embedding, without variational inference.

Table 2 shows a comparison of the detection performance. As for the *micro F1* evaluation metric, our model achieves the highest performance (83.54%) on the FNC-1 testing subset. The average method can lose emphasis or key word information in a claim; the CNN-based method can only capture local dependency among the text with limit to the filter size; the RNN-based method can obtain semantic relationship in a sequential manner. Differently, the self-attention method is able to combine embedding information between each pair of words, which means more accurate semantic matching of the claim and the piece of text. Compared with the deterministic self-attention, our method is a stochastic approach that is experimentally proven to better integrate the vector embedding of each word.

[1] `https://github.com/FakeNewsChallenge/fnc-1`

4 Conclusion

We propose a variational self-attention model (VSAM) that builds a self-attention vector as random variables by imposing a probabilistic distribution. Compared with conventional deterministic counterpart, the stochastic units incorporated by VSAM allow multi-modal attention distributions. Furthermore, by marginalizing over the latent variables, VSAM is more robust against overfitting, which is important for small datasets. Experiments on the stance detection task demonstrate the superiority of our method.

Acknowledgments

This project was funded by the EPSRC Fellowship titled "Task Based Information Retrieval", grant reference number EP/P024289/1.

References

[1] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014.

[2] H. Bahuleyan, L. Mou, O. Vechtomova, and P. Poupart. Variational attention for sequence-to-sequence models. In *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 1672–1682, 2018.

[3] Z. Lin, M. Feng, C. N. dos Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio. A structured self-attentive sentence embedding. *CoRR*, abs/1703.03130, 2017.

[4] R. Luo, W. Zhang, X. Xu, and J. Wang. A neural stochastic volatility model. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, Louisiana, USA, February 2-7, 2018*, 2018.

[5] Y. Miao, L. Yu, and P. Blunsom. Neural variational inference for text processing. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, pages 1727–1736, 2016.

[6] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, pages 3111–3119, 2013.

[7] A. P. Parikh, O. Täckström, D. Das, and J. Uszkoreit. A decomposable attention model for natural language inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 2249–2255, 2016.

[8] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1532–1543, 2014.

[9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 6000–6010, 2017.

[10] Q. Zhang, E. Yilmaz, and S. Liang. Ranking-based method for news stance detection. In *Companion of the The Web Conference 2018 on The Web Conference 2018*, pages 41–42. International World Wide Web Conferences Steering Committee, 2018.