An Ordered Semantic Embedding for Sentence

Yi Zhang
School of Electronics Engineering and Computer Science, Peking University
{zhangyi16}@pku.edu.cn

1 Abstract

In this paper, we propose a method to extract a sentence embedding which reflects different aspects of semantics in a weak order. Most existing methods typically attempt to capture the whole semantics of a sentence with a weighted sum of word embeddings. Much information may be lost in the process. More elaborate methods try to extract different aspects of semantics, ignoring their semantic connections. However, this treatment is counter-intuitive. Human read sentences efficiently by paying attention to different parts of a sentence orderly. These semantic aspects have their inner relationship or logic. Inspired by this, we propose to extract the semantics orderly and represent a single aspect with a vector. The final sentence embedding will be a matrix that reflects the main semantic aspects and also keeps their inner logic.

2 Introduction

Many NLP tasks rely on the meaningful distributed representations of individual words, also known as word embeddings (Mikolov et al., 2013). Efforts to obtain embeddings for larger chunks of text, such as sentences, have not been so successful. Since understanding the meaning of a sentence is a prerequisite for many natural language processing tasks, much remains to be done to obtain satisfying representations of phrases and sentences. Related methods generally fall into two categories. The first category concerns universal sentence embeddings and are usually trained by unsupervised learning (Bontcheva et al., 2009). While the other category concerns models and are trained specifically for a certain task. (Lin et al., 2017).

Most of the existing models typically compress the whole sentence information into a vector. For example, take the final hidden state of a RNN or the max (or average) pooling over RNN hidden states as the sentence embedding. Semantic information like dependency tree may be used to improve the representations (Mou et al., 2015; Tai et al., 2015). Some researches propose to extract the sentence information based on attention mechanisms (Wang et al., 2016). Such that the sentence embedding may involve the important information of a sentence. Recently, a self-attention (Lin et al., 2017) mechanism is proposed to extract different aspects of the sentence. The final sentence embedding will be encoded into multiple vector representations.

However, these methods either focus on single aspect of a sentence or try to capture multiple aspects ignoring the inner semantic relationship of a sentence. Thus, we propose to extract different semantic aspects in a more natural way like human reading. After encoding the sentence with a RNN, our decoder generates a semantic aspect by a simple attention mechanism at each time step. Unlike previous work, where all important aspects are searched in the whole sentence, we feed the network with the varying context information that has subtracted former concerns at each time step. A new aspect is generated based on the varying context information. Thus we can find all aspects in a sentence along the time step and the next aspect is searched in the remaining sentence, just like human reading. The final sentence embedding contains different semantic aspects and their inner connections are also preserved.

3 Related Work

The methods of sentence representation generally fall into two categories. The first category learns the universal sentence embeddings. These embeddings are usually trained by unsupervised learning (Hill et al., 2016). This includes SkipThought vectors (Kiros et al., 2015), ParagraphVector (Le and Mikolov, 2014), recursive auto-encoders citesocher2011semi,socher2013recursive, Sequential Denoising Autoencoders, FastSent (Hill et al., 2016), etc.The other category learns the embeddings for a certain task. These models are usually trained by supervised learning. One generally finds that specifically trained sentence embeddings perform better than generic ones, although generic ones can be used in a semi-supervised setting, exploiting large unlabeled corpora.

Recently, neural networks are used in a wide range of tasks, including named entity recognition (He and Sun, 2017; Sun and He, 2017), word segmentation (Xu et al., 2017) and summarization (Ma et al., 2017). Most of these work encode the sentence information with neural networks, such as recurrent networks (Hochreiter and Schmidhuber, 1997), recursive networks (Socher et al., 2013) and convolutional networks (Kalchbrenner et al., 2014; Kim, 2014). A simple and common approach in these methods is using the final hidden state of the RNN or using the the max (or average) pooling results over CNNs to represent sentences. These representations then can be used to solve a wide variety of tasks including classification and ranking. Additional works have also been done in exploiting the efficiency of neural networks such as sparse propagation (Sun et al., 2017; Wei et al., 2017).

Our work is also related to the sequence-to-sequence model (Cho et al., 2014), one of the most successful generative neural model. It is widely applied in machine translation and text summarization. Our decoder part is based on neural attention mechanism (Bahdanau et al., 2014). There are many other methods to improve neural attention model (Jean et al., 2015; Luong et al., 2015).

4 Approach

Our proposed model consists of two parts. First, a bi-directional recurrent neural network is used to encode the sentence. The hidden state at each time step contains both past and future information. Thus the averaged hidden states can be seen as the original context vector. Second, we use another recurrent neural network, which acts like a decoder, to find an important semantic aspect at every time step. This network is fed with the context vector that has subtracted former concerned information. If we combine the vector along the time step, we will obtain an embedding matrix M of the sentence, where each row reflects a specific aspect of the sentence orderly. To use the internal sequential information of these vectors, a recurrent neural network can be naturally used. We will introduce these two parts separately and give some alternatives to subtract former information.

4.1 Bi-LSTM

For a sequence of T words $(w_1, w_2, ..., w_T)$, the network computes a set of T hidden representations $h_1, ..., h_T$, where h_T is the concatenation of a forward LSTM and a back ward LSTM that read the sentence in two opposite directions.

$$\overrightarrow{h_t} = \overrightarrow{LSTM_t}(w_1, w_2, ..., w_T)$$

$$\overleftarrow{h_t} = \overleftarrow{LSTM_t}(w_1, w_2, ..., w_T)$$

$$h_t = [\overrightarrow{h_t}, \overleftarrow{h_t}]$$

A sentence is represented by the averaged hidden states. This averaged vector is considered to have included all information in the sentence and will be used as the first context vector c_1.

4.2 Ordered Attention Decoder

We design a decoder which imitates the reading process of human. At every time step, the model takes a context vector as the input and outputs a distribution over the hidden states. A certain aspect of the sentence is the weighted sum of hidden states computed by the distribution. The context vector is a varying vector that has subtracted the information we have focused on. Just like that human will concentrate on different aspects orderly of the remaining text.

We use a recurrent neural network to implement the functions. Unlike traditional sequence-to-sequence models which take the output of last step as the current input, our model do not directly use the output of last step as input. The outputs of decoder, which are the semantic aspects we want to capture, are tightly associated with the contextual information but they can not decide each other. Thus we use the varying context vector as input at each time step. Such treatment results in two advantages. First, each semantic aspect is searched in a narrowed range, which alleviates the redundancy among these aspects. Second, these aspects will be generated in a weak order. Their inner logic is preserved. This process imitates the human reading behavior and gives a better interpretation of sentence embedding.

More formally, given the context c_i, the model outputs a vector of weights α over the whole LSTM hidden states H.

$$\alpha^i = softmax(H \cdot f(W, c_i))$$

$$m_i = \sum_{i=1}^{T} h_t \cdot \alpha_t^i$$

Here W is the parameter matrix and f is the recurrent neural network.

4.3 The Varying Contextual Vector

We explore several ways to subtract concerned information from the context vector.

The simplest way is directly subtracting the concerned information from the context vector.

$$\hat{c_i} = c_i - m_i$$

The second way is to make the current vector c_i orthogonal to the context vector at $i-1$.

$$\hat{c_i} = c_i - \frac{c_i^T \hat{c_{i-1}}}{\hat{c_{i-1}}^T \hat{c_{i-1}}} \hat{c_{i-1}}$$

4.4 Penalization Term

Since our model uses the attention mechanism over the encoder hidden states along every time step, we adapt the penalization term proposed by Lin et al. (2017) to prevent the redundancy problems if the attention mechanism always provides similar summation weights.

The dot product of A and its transpose, subtracted by an identity matrix, is regarded as a measure of redundancy:

$$P = \| (AA^T - I) \|_2^T$$

Where $\| \quad \|$ stands for the Frobenius norm of a matrix. Similar to adding an L2 regularization term, this penalization term P will be multiplied by a coefficient, and we minimize it together with the original loss, which is dependent on the downstream application.

The diagonal elements is the squared results of an attention distribution a. We subtract an identity matrix from AA^T so that forces the elements on the diagonal of AA^T to approximate 1, which encourages each summation vector a to focus on as few number of words as possible, forcing each vector to be focused on a single aspect, and all other elements to 0, which punishes redundancy between different summation vectors.

5 Conclusion

In this paper, we propose a method to extract a sentence embedding which reflects different aspects of semantics in a weak order. Unlike previous work that using the final hidden state of the RNN or using the the max (or average) pooling results over CNNs to represent sentences, our method propose to extract the semantics orderly and represent a single aspect with a vector. At every time step, an attention mechanism is used to capture a semantic aspect of the sentence with a varying context vector that has subtracted concerned information. The final sentence embedding will be a matrix that reflects the main semantic aspects and also keeps their inner logic.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR* abs/1409.0473.

Kalina Bontcheva, Brian Davis, Adam Funk, Yaoyong Li, and Ting Wang. 2009. *Human language technologies*. Springer.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014*. pages 1724–1734.

Hangfeng He and Xu Sun. 2017. A unified model for cross-domain and semi-supervised named entity recognition in chinese social media. In *AAAI*. pages 3216–3222.

Felix Hill, Kyunghyun Cho, and Anna Korhonen. 2016. Learning distributed representations of sentences from unlabelled data. *arXiv preprint arXiv:1602.03483* .

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.

Sébastien Jean, KyungHyun Cho, Roland Memisevic, and Yoshua Bengio. 2015. On using very large target vocabulary for neural machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics, ACL 2015*. pages 1–10.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188* .

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882* .

Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *Advances in neural information processing systems*. pages 3294–3302.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*. pages 1188–1196.

Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding. *arXiv preprint arXiv:1703.03130* .

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015*. pages 1412–1421.

Shuming Ma, Xu Sun, Jingjing Xu, Houfeng Wang, Wenjie Li, and Qi Su. 2017. Improving semantic relevance for sequence-to-sequence learning of chinese social media text summarization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 2: Short Papers*. pages 635–640.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* .

Lili Mou, Hao Peng, Ge Li, Yan Xu, Lu Zhang, and Zhi Jin. 2015. Discriminative neural sentence modeling by tree-based convolution. *arXiv preprint arXiv:1504.01106* .

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*. pages 1631–1642.

Xu Sun and Hangfeng He. 2017. F-score driven max margin neural network for named entity recognition in chinese social media. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 2: Short Papers*. pages 713–718.

Xu Sun, Xuancheng Ren, Shuming Ma, and Houfeng Wang. 2017. meprop: Sparsified back propagation for accelerated deep learning with reduced overfitting. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*. pages 3299–3308.

Kai Sheng Tai, Richard Socher, and Christopher D Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. *arXiv preprint arXiv:1503.00075* .

Shaonan Wang, Jiajun Zhang, and Chengqing Zong. 2016. Learning sentence representation with guidance of human attention. *arXiv preprint arXiv:1609.09189* .

Bingzhen Wei, Xu Sun, Xuancheng Ren, and Jingjing Xu. 2017. Minimal effort back propagation for convolutional neural networks .

Jingjing Xu, Shuming Ma, Yi Zhang, Bingzhen Wei, Xiaoyan Cai, and Xu Sun. 2017. Transfer deep learning for low-resource chinese word segmentation with a novel neural network. In *Natural Language Processing and Chinese Computing - 6th CCF International Conference, NLPCC*

2017, Dalian, China, November 8-12, 2017, Proceedings. pages 721–730.