Character-level Intra Attention Network for Natural Language Inference

Han Yang and **Marta R. Costa-jussà** and **José A. R. Fonollosa**
TALP Research Center
Universitat Politècnica de Catalunya
han.yang@est.fib.upc.edu {marta.ruiz,jose.fonollosa}@upc.edu

Abstract

Natural language inference (NLI) is a central problem in language understanding. End-to-end artificial neural networks have reached state-of-the-art performance in NLI field recently.

In this paper, we propose Character-level Intra Attention Network (CIAN) for the NLI task. In our model, we use the character-level convolutional network to replace the standard word embedding layer, and we use the intra attention to capture the intra-sentence semantics. The proposed CIAN model provides improved results based on a newly published MNLI corpus.

1 Introduction

Natural language inference in natural language processing refers to the problem of determining a directional relation between two text fragments. Given a sentence pair (premise, hypothesis), the task is to predict whether hypothesis is entailed by premise, hypothesis is contradicted to premise, or whether the relation between premise and hypothesis is neutral.

Recently, the dominating trend of works in natural language processing is based on artificial neural networks, which aims at building deep and complex encoder to transform a sentence into encoded vectors. For instance, there are recurrent neural network (RNN) based encoders, which recursively concatenate each word with its previous memory, until the whole information of a sentence has been derived. The most common RNN encoders are Long Short-Term Memory Networks (LSTM; Hochreiter and Schmidhuber, 1997) and Gated Recurrent Unit (Chung et al., 2014). RNNs have surpassed the performance of traditional baselines in many NLP tasks (Dai and Le, 2015). There are also convolutional neural network (CNN; LeCun et al., 1989) based encoders, which concatenate the sentence information by applying multiple convolving filters over the sentence. CNNs have achieved state-of-the-art results on various NLP tasks (Collobert et al., 2011).

To evaluate the quality of the NLI model, the Stanford Natural Language Inference (SNLI; Bowman et al., 2015) corpus of 570K sentence pairs was introduced. It serves as a standard benchmark for NLI task. However, most of the sentences in SNLI corpus are short and simple, which limit the room for fine-grained comparisons between models. Currently, a more comprehensive Multi-Genre NLI corpus (MNLI; Williams et al., 2017) of 433K sentence pairs was released, aiming at evaluating large-scale NLI models. Authors gave out some baseline results accompanied by the publish of MNLI corpus, the BiLSTM model achieves an accuracy of 67.5, and the Enhanced Sequential Inference Model (Chen et al., 2016) achieves an accuracy of 72.4.

Among those encoders for NLI task, most of them use word-level embedding, and initialize the weight of the embedding layer with pre-trained word vectors such as GloVe (Pennington et al., 2014). The pre-trained word vectors helps the encoders to catch richer semantic information. However, it also has its downside. As the growth of vocabulary size in the modern corpus, there will be more and more out-of-vocabulary (OOV) words that are not presented in the pre-trained word embedding vector. As the word-level embedding is blind to subword information (e.g. morphemes), it leads to high perplexities for those OOV words.

In this paper, we use the BiLSTM model from (Williams et al., 2017) as the baseline model for the evaluation of the MNLI corpus. To augment the baseline model, firstly, a character-level

Proceedings of the 2nd Workshop on Evaluating Vector-Space Representations for NLP, pages 46–50,
Copenhagen, Denmark, September 7–11, 2017.

convolutional neural network (CharCNN; Kim et al., 2016) is applied. We use the CharCNN to replace the word embedding layer in the baseline model, which will be computed from the characters of corresponding word. Secondly, the intra attention mechanism introduced by (Yang et al., 2016) will be applied, to enhance the model with a richer information of substructures of a sentence.

2 Model Development

2.1 BiLSTM Baseline

The baseline model we used here is introduced by (Williams et al., 2017) accompanied with the publication of MNLI corpus. It has a 5-layer structure which is shown in Figure 1.

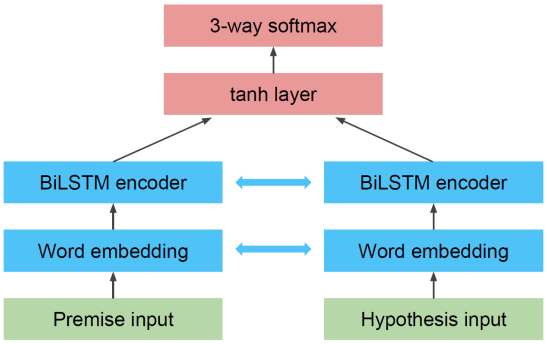

Figure 1: BiLSTM model architecture

In the baseline model, a word embedding layer initialized with pre-trained GloVe vectors (840B token version) is implemented to transform the input text into sequence of word vectors. OOV words are initialized randomly. Then, the sentence representation vector h is produced by implementing an average pooling over the BiLSTM hidden states $[h_0, h_1, \cdots, h_n]$. Finally, the concatenation of encoded premise and hypothesis representation vector is passed through a tanh layer followed by a three-way softmax classifier to attain the label prediction.

2.2 Character-level Convolutional Neural Network

In the baseline model, the input x_t to the BiLSTM encoder layer at time t is sequence of pre-trained word embeddings. Those pre-trained word embeddings can boost the performance of the model. However, it is limited to the finite-size of vocabulary. Here we replace the word embedding layer with a character-level convolutional neural network (CharCNN; Kim et al., 2016) for language modeling, which also achieved success in machine translation (Costa-Jussà and Fonollosa, 2016).

We define the text sentence input as vector $C^k \in R^{d \times l}$, where $k \in K$ is the k-th word in a sentence, d is the dimensionality of character embeddings, l is the length of characters in k-th word. Then a set of narrow convolutions between C^k and filter H is applied, followed with a max-over-time (max pooling) as shown in Equation 1-2.

$$f^k[i] = \tanh(\langle C^k[*, i : i + \omega - 1], H \rangle + b) \quad (1)$$

$$y^k = \max_i f^k[i] \quad (2)$$

The concatenation of those max pooling values y^k provides us with a representation vector y of each sentence. Then, a highway network is applied upon y, as shown in Equation 3, where g is a non-linear transformation, $t = \sigma(W_T y + b_T)$ is called the transform gate, and $(1 - t)$ is called the carry gate. Highway layers allow for training of deep networks by adaptively carrying some dimensions of the input y directly to the output z.

$$z = t \odot g(W_H y + b_H) + (1 - t) \odot y \quad (3)$$

Experiment conducted by (Kim et al., 2016) has shown that the CNN layer can extract the orthographic features of words (e.g. *German* and *Germany*). It has also been shown that highway layer is able to encode semantic features that are not discernable from orthography alone. For instance, after highway layers the nearest neighbor word of *you* is *we*, which is orthographically distinct from *you*.

2.3 Intra Attention Mechanism

In the baseline model, the BiLSTM encoder takes an average pooling over all its hidden states to produce a single representation vector of each sentence. However, this has its bottleneck as we intuitively know that not all words (hidden states) contribute equally to the sentence representation. To augment the performance of RNN based encoder, the concept of attention mechanism was introduced by (Bahdanau et al., 2014) for machine translation. Attention mechanism is a hidden layer which computes a categorical distribution to make a soft-selection over source elements (Kim et al., 2017). It has recently demonstrated success on tasks such as parsing text (Vinyals et al., 2015), sentence summarization (Rush et al., 2015) and

also on a wide range of NLP tasks (Cheng et al., 2016).

Here we implemented the Intra Attention mechanism introduced by (Yang et al., 2016) for document classification. We define the hidden states as the output of the BiLSTM encoder as $h_t \in [h_0, h_1, \cdots, h_n]$, the intra attention is applied upon the hidden states to get the sentence representation vector h, specifically,

$$u_t = \tanh(W_\omega h_t + b_\omega) \tag{4}$$

$$\alpha_t = \frac{exp(u_t^T u_\omega)}{\sum_t exp(u_t^T u_\omega)} \tag{5}$$

$$h = \sum_t \alpha_t h_t \tag{6}$$

It first feed all hidden states h_t through a nonlinearity to get u_t as the hidden representation of h_t. Then it uses a $softmax$ function to catch the normalized importance weight matrix α_t. After that, the sentence representation vector h is computed by a weighted sum of all hidden states h_t with the weight matrix α_t. The context vector u_ω can be seen as a high-level representation of the importance of informative words.

2.4 Character-level Intra Attention Network

The overall architecture of the Character-level Intra Attention Network (CIAN) is shown in Figure 2. The CIAN model is consisted with 7 layers, of which the first and the last layers are the same with our baseline model. The 4 layers in middle are our augmented layers that has been introduced in this section.

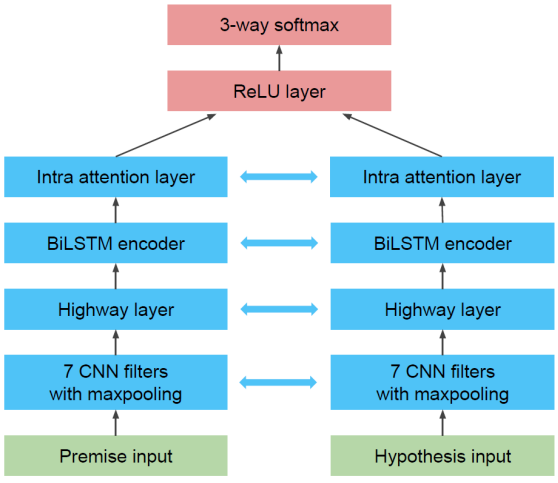

Figure 2: CIAN model architecture

The input text is firstly set to lowercase, then it is vectorized according to the tokenization list [abcdefghijklmnopqrstuvwxyz0123456789,;.!?:'"()[]{}]. Those characters not in the list are initialized with a vector of zero. After that we use 7 filters in CIAN model's CNN layer. The widths of the CNN filters are $w = [1, 2, 3, 4, 5, 6, 7]$, and the corresponding filters' size are $[\min\{200, 50 \cdot w\}]$. Two highway layers are implemented following the CNN layer. The attention layer uses weighted sum of all hidden states h_t with the attention weight matrix α_t to encode each sentence into a fixed-length sentence representation vector. Finally a ReLU layer and a three-way softmax classifier use those representation vectors to conduct the prediction.

3 Experiments

3.1 Data

We evaluated our approach on the Multi-Genre NLI (MNLI) corpus, as a shared task for RepEval 2017 workshop (Nangia et al., 2017). We train our CIAN model on a mixture of MNLI and SNLI corpus, by using a full MNLI training set and a randomly selected 20 percent of the SNLI training set at each epoch.

3.2 Hyper Parameters

The BiLSTM encoder layer use 300D hidden states, thus 600D as its a bidirectional encoder. Dropout (Srivastava et al., 2014) is implemented with a dropout rate of 0.2 to prevent the model from overfitting. Parameter weights for premise encoder and hypothesis encoder are shared using siamese architecture. The Adam optimizer (Kingma and Ba, 2014) is used for training with backpropagation.

The model has been implemented using Keras and we have released the code [1]. The training took approximately one hour for one epoch on GeForce GTX TITAN, and we stopped training after 40 epochs as an early stopping regularization.

3.3 Result

We compared the results of CIAN model with the results of BiLSTM model given by (Williams et al., 2017). Table 1 shows that the accuracy is improved by 0.9 percent in matched test set, and 0.6 percent in mismatched test set.

[1] https://github.com/yanghanxy/CIAN

Model	Matched	Mismatched
BiLSTM	67.0	67.6
CIAN	67.9	68.2

Table 1: Test set accuracies (%) on MNLI corpus.

	Matched		Mismatched	
Tag	**BiLSTM**	**CIAN**	**BiLSTM**	**CIAN**
CONDITIONAL	100	48	100	62
WORD_OVERLAP	50	79	57	62
NEGATION	71	71	69	70
ANTO	67	82	58	70
LONG_SENTENCE	50	68	55	63
TENCE_DIFFERNCE	64	65	71	72
ACTIVE/PASSIVE	75	87	82	90
PARAPHRASE	78	88	81	89
QUANTITY/TIME	50	47	46	44
COREF	84	67	80	72
QUANTIFIER	64	63	70	69
MODAL	66	66	64	70
BELIEF	74	71	73	70

Table 2: Accuracies (%) on matched and mismatched expert-tagged development data.

We conducted error analysis based on expert-tagged development data released by the organizers of RepEval 2017 shared task. The results are shown in Table 2. From the results, it can be seen that the accuracy for WORD_OVERLAP, LONG_SENTENCE, ACTIVE/PASSIVE and PARAPHRASE have been improved significantly in both matched and mismatched development set. While the accuracy for CONDITIONAL and COREF haven been decreased in both development set.

We also conducted visualization on the attention weights α_t of the intra attention layer. By doing so, we we can understand how the model judges the NLI relation between two sentences.

Figure 3 is visualizations of attention weights for 2 sentence pairs, with premise at left and hypothesis at right. Each word is attained with a color block. The darker the color, the greater the attention weight, which means the higher importance contributed to the sentence representation.

From the Visualization, it could be seen that the model has more attention on words with similar semantic meaning (e.g. *love* and *enjoy*), and the model applies more attention on overlapped words (e.g. *efficiencies* and *efficiencies*).

4 Conclusion

In this paper, we presented a Character-level Intra Attention Network (CIAN) for the task of natural language inference. Experimental results

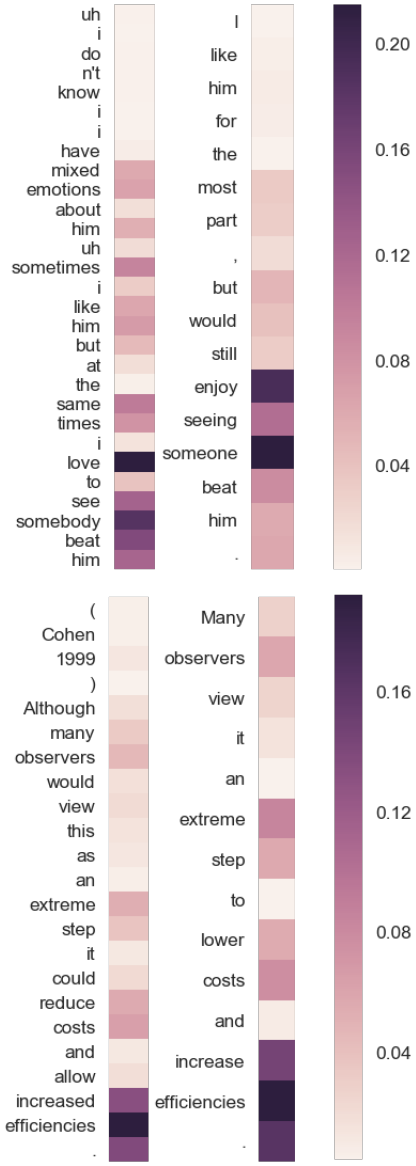

Figure 3: Visualization of attention weights of sentence pair 254941e (top) and 192997e (bottom)

demonstrate that our model slightly outperforms the baseline model upon the MultiNLI corpus. The CharCNN layers helps the model to capture rich semantic and orthographic features. The intra attention layer augment the model's ability to efficiently encode long sentences, and it enhances the models' interpretability by visualizing the attention weights.

In general, the model presented in this paper is a sequence encoder that do not need any specific pre-processing or outside data like pre-trained word embeddings. Thus, it can be easily applied to other autoencoder architecture tasks such as language modeling, sentiment analysis and question answering.

Acknowledgments

This work is supported by the Spanish Ministerio de Economía y Competitividad and Fondo Europeo de Desarrollo Regional through contract TEC2015-69266-P (MINECO/FEDER, UE), by the postdoctoral senior grant Ramón y Cajal, and by the China Scholarship Council (CSC) under grant No.201506890038.

References

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

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015*. pages 632–642.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, and Hui Jiang. 2016. Enhancing and combining sequential and tree LSTM for natural language inference. *CoRR* abs/1609.06038.

Jianpeng Cheng, Li Dong, and Mirella Lapata. 2016. Long short-term memory-networks for machine reading. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016*. pages 551–561.

Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR* abs/1412.3555.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel P. Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research* 12:2493–2537.

Marta R. Costa-Jussà and José A. R. Fonollosa. 2016. Character-based neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, Volume 2: Short Papers*.

Andrew M. Dai and Quoc V. Le. 2015. Semi-supervised sequence learning. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015*. pages 3079–3087.

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

Yoon Kim, Carl Denton, Luong Hoang, and Alexander M. Rush. 2017. Structured attention networks. *CoRR* abs/1702.00887.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M. Rush. 2016. Character-aware neural language models. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*. pages 2741–2749.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR* abs/1412.6980.

Yann LeCun, John S. Denker, and Sara A. Solla. 1989. Optimal brain damage. In *Advances in Neural Information Processing Systems 2, [NIPS Conference, 1989]*. pages 598–605.

Nikita Nangia, Adina Williams, Angeliki Lazaridou, and Samuel R. Bowman. 2017. The repeval 2017 shared task: Multi-genre natural language inference with sentence representations. In *Proceedings of RepEval 2017: The Second Workshop on Evaluating Vector Space Representations for NLP*. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, A meeting of SIGDAT, a Special Interest Group of the ACL*. pages 1532–1543.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015*. pages 379–389.

Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15(1):1929–1958.

Oriol Vinyals, Lukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey E. Hinton. 2015. Grammar as a foreign language. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015*. pages 2773–2781.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2017. A broad-coverage challenge corpus for sentence understanding through inference. *CoRR* abs/1704.05426.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alexander J. Smola, and Eduard H. Hovy. 2016. Hierarchical attention networks for document classification. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. pages 1480–1489.