Refining Raw Sentence Representations for Textual Entailment Recognition via Attention

Jorge A. Balazs, Edison Marrese-Taylor, Pablo Loyola, Yutaka Matsuo
University of Tokyo, Graduate School of Engineering, Japan
{jorge, emarrese, pablo, matsuo}@weblab.t.u-tokyo.ac.jp

Abstract

In this paper we present the model used by the team Rivercorners for the 2017 RepEval shared task. First, our model separately encodes a pair of sentences into variable-length representations by using a bidirectional LSTM. Later, it creates fixed-length raw representations by means of simple aggregation functions, which are then refined using an attention mechanism. Finally it combines the refined representations of both sentences into a single vector to be used for classification. With this model we obtained test accuracies of 72.057% and 72.055% in the matched and mismatched evaluation tracks respectively, outperforming the LSTM baseline, and obtaining performances similar to a model that relies on shared information between sentences (ESIM). When using an ensemble both accuracies increased to 72.247% and 72.827% respectively.

Introduction

The Second Workshop on Evaluating Vector Space Representations for NLP (RepEval 2017) features a shared task meant to evaluate natural language understanding models based on sentence encoders by the means of NLI in the style of a three-class balanced classification problem over sentence pairs. The shared task includes two evaluations, a standard in-domain (matched) evaluation in which the training and test data are drawn from the same sources, and a cross-domain (mismatched) evaluation in which the training and test data differ substantially. This cross-domain evaluation is aimed at testing the ability of submitted systems to learn representations of sentence meaning that capture broadly useful features.

Model Description

General Architecture

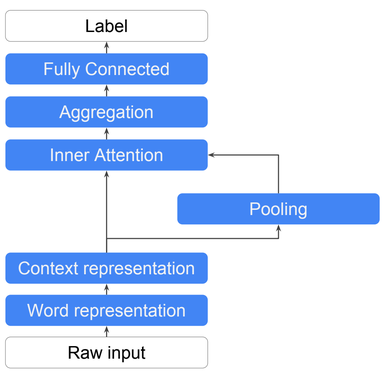

Figure 1: General architecture of our proposed model.

Word Representation Layer

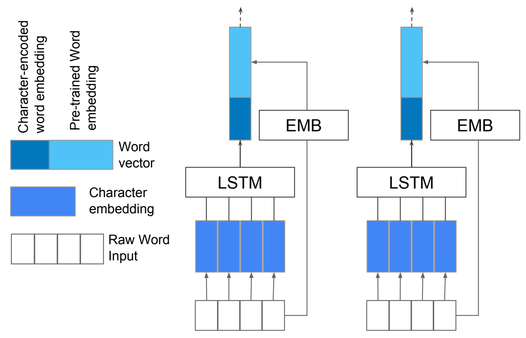

Figure 2: Word representation layer architecture. For each word we embed each of its characters, we feed the resulting vectors to a LSTM, take the last hidden state and conatenate it with a pretrained word embedding.

Context Representation Layer

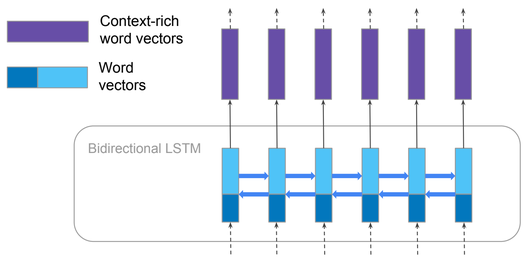

Figure 3: Context representation layer architecture. Word vectors returned by the previous layer are fed to a bidirectional LSTM to generate context-rich word representations.

Pooling and Attention Layers

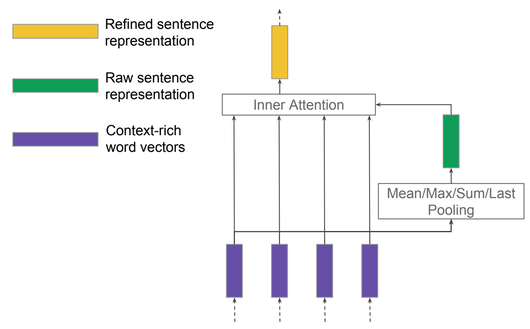

Figure 4: Pooling and attention layers. A raw sentence representation is created by pooling the context-rich word vectors returned by the previous layer. Then an attention mechanism uses both this newly created raw representation and the previous context-rich vectors to create a more refined sentence representation.

$$\boldsymbol{u}_i = \boldsymbol{v}^\top \tanh(W[\bar{\boldsymbol{h}}; \boldsymbol{h}_i]) \tag{1}$$

$$\boldsymbol{\alpha}_i = \frac{\exp \boldsymbol{u}_i}{\sum_{k=1}^{n} \exp \boldsymbol{u}_k} \tag{2}$$

$$\bar{\boldsymbol{h}}' = \sum_{i=1}^{n} \boldsymbol{\alpha}_i \boldsymbol{h}_i \tag{3}$$

Aggregation Layer

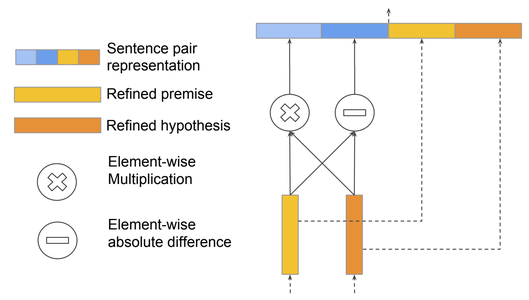

Figure 5: Aggregation layer architecture. Premise and hypothesis refined vectors are multiplied and subtracted, and the results are concatenated to the original refined vectors to obtain the final sentence representation.

$$\boldsymbol{h}_{mul} = \bar{\boldsymbol{h}}'_P \odot \bar{\boldsymbol{h}}'_H \tag{4}$$

$$\boldsymbol{h}_{dif} = |\bar{\boldsymbol{h}}'_P - \bar{\boldsymbol{h}}'_H| \tag{5}$$

$$\boldsymbol{r} = [\bar{\boldsymbol{h}}'_P; \bar{\boldsymbol{h}}'_H; \boldsymbol{h}_{mul}; \boldsymbol{h}_{dif}] \tag{6}$$

Fully Connected Layer

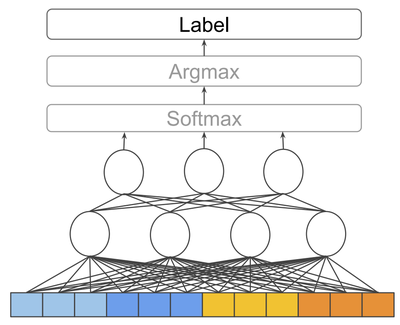

Figure 6: Fully connected layer architecture. The final sentence representation is fed to a softmax function and the greatest probability's index is chosen as the predicted label.

Experimental Setup

Corpus: MultiNLI + 15% SNLI randomly sampled once

Word Embeddings: GloVe 300d 840b, not fine tuned

Character Embeddings: $Uniform(-0.05, 0.05)$, 20d

Character-level LSTM output dim.: 50

Word-level BiLSTM output dim.: 300×2

Attention: W: 1200×600 matrix; $\boldsymbol{v}$: vector of dim 600. Both initialized from $Uniform(-0.005, 0.005)$

Fully-connected Layer: 3-layer MLP with 2000 hidden units each and ReLU activations

Optimizer: RMSprop, learning rate 0.001

Dropout: 0.25 only applied between layers of the MLP

Miscellaneous: All characters were made lowercase, numbers transformed into single token, ignored sentence pairs with premises longer than 200 words, and those with an undefined label ("-")

Results

Method	w/o. chars	w. chars
mean	71.3 ± 1.2	71.3 ± 0.7
sum	70.7 ± 1.0	70.9 ± 0.8
last	70.9 ± 0.6	71.0 ± 1.2
max	70.6 ± 1.1	71.0 ± 1.1

Table 1: Mean matched validation accuracies (%) broken down by type of pooling method and presence or absence of character embeddings. Confidence intervals are calculated at 95% confidence over 10 runs for each method.

Method	w/o. chars	w. chars
mean	**72.3**	71.8
sum	71.6	71.6
last	71.4	**72.1**
max	71.1	71.6

Table 2: Best matched validation accuracies (%) obtained by each pooling method in presence and absence of character embeddings.

Genre	CBOW	ESIM	InnerAtt
Fiction	67.5	73.0	73.2
Government	67.5	74.8	75.2
Slate	60.6	67.9	67.2
Telephone	63.7	72.2	73.0
Travel	64.6	73.7	72.8
9/11	63.2	71.9	70.5
Face-to-face	66.3	71.2	74.5
Letters	68.3	74.7	75.4
Oup	62.8	71.7	71.5
Verbatim	62.7	71.9	69.5
MultiNLI Overall	**64.7**	**72.2**	**72.3**

Table 3: Validation accuracies (%) for our best model broken down by genre. Both CBOW and ESIM results are reported as in [11].

Conclusions and Future Work

We presented the model used by the team Rivercorners in the 2017 RepEval shared task. Despite being conceptually simple and not relying on shared information between premise and hypothesis for encoding each sentence, nor on tree structures, our implementation achieved results as good as the ESIM model.

As future work we plan to incorporate part-of-speech embeddings to our implementation and concatenate them at the same level as we did with the character embeddings. We also plan to use pretrained character embeddings to see whether they have any positive impact on performance.

Additionally, we think we could obtain better results by fine-tuning some hyperparameters such as the character embedding dimensions, the character-level LSTM encoder output dimension, and the Dense Layer architecture.

Further, we would like to see how different types of attention affect the overall performance. For this implementation we used the *concat* scoring scheme (eq. 1), as described by Luong et al. [6], but there are several others that could provide better results.

Finally, we would like to exploit the structured nature of dependency parse trees by means of recursive neural networks [9] to enrich our initial sentence representations.

References

[1] Bowman, S. R., Angeli, G., Potts, C., and Manning, C. D. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal, September 2015. Association for Computational Linguistics. URL http://aclweb.org/anthology/D15-1075.

[2] Chen, Q., Zhu, X., Ling, Z.-H., Wei, S., Jiang, H., and Inkpen, D. Enhanced lstm for natural language inference. In *Proc. ACL*, 2017.

[3] Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. URL http://www.bioinf.jku.at/publications/older/2604.pdf.

[4] Lin, Z., Feng, M., dos Santos, C. N., Yu, M., Xiang, B., Zhou, B., and Bengio, Y. A structured self-attentive sentence embedding. In *Proceedings of the 5th International Conference on Learning Representations (ICLR)*, 2017. URL https://arxiv.org/pdf/1703.03130.pdf.

[5] Liu, Y., Sun, C., Lin, L., and Wang, X. Learning natural language inference using bidirectional LSTM model and inner-attention. *CoRR*, abs/1605.09090, 2016. URL http://arxiv.org/pdf/1605.09090.pdf.

[6] Luong, T., Pham, H., and Manning, C. D. Effective Approaches to Attention-based Neural Machine Translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal, sep 2015. Association for Computational Linguistics. URL http://aclweb.org/anthology/D15-1166.

[7] Mou, L., Peng, H., Li, G., Xu, Y., Zhang, L., and Jin, Z. Discriminative neural sentence modeling by tree-based convolution. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2315–2325, Lisbon, Portugal, September 2015. Association for Computational Linguistics. URL http://aclweb.org/anthology/D15-1279.

[8] Pennington, J., Socher, R., and Manning, C. D. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014. URL http://www.aclweb.org/anthology/D14-1162.

[9] Tai, K. S., Socher, R., and Manning, C. D. Improved semantic representations from tree-structured long short-term memory networks. *arXiv preprint arXiv:1503.00075*, 2015. URL https://arxiv.org/pdf/1503.00075.pdf.

[10] Wang, Z., Hamza, W., and Florian, R. Bilateral multi-perspective matching for natural language sentences. *CoRR*, abs/1702.03814, 2017. URL http://arxiv.org/abs/1702.03814.

[11] Williams, A., Nangia, N., and Bowman, S. R. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*, 2017. URL http://arxiv.org/pdf/1704.05426.pdf.

[12] Xiong, C., Zhong, V., and Socher, R. Dynamic coattention networks for question answering. *arXiv preprint arXiv:1611.01604*, 2016.