Neural Machine Translation By Jointly Learning To Align and Translate

Dzmitry Bahdanau, KyungHyun Cho, and Yoshua Bengio

International Conference on Learning Representations, 2015

Problems Addressed

1. The use of a fixed-length vector is a bottleneck in improving the performance of the basic encoder–decoder architecture.

2. A neural network needs to be able to compress all the necessary information of a source sentence into a fixed-length vector, which makes it difficult for the neural network to cope with long sentences, especially those that are longer than the sentences in the training corpus.

Proposed Solution

1. The proposed solution is to allow a model to automatically (soft-)search for parts of a source sentence that are relevant to predicting a target word, without having to form these parts as a hard segment explicitly.

2. The most important distinguishing feature of this approach from the basic encoder–decoder is that it does not attempt to encode a whole input sentence into a single fixed-length vector. Instead, it encodes the input sentence into a sequence of vectors and chooses a subset of these vectors adaptively while decoding the translation.

3. The architecture consists of a bidirectional RNN as the encoder and a simple RNN as the decoder.

Encoder

- The encoder maps the input sentence to a sequence of *annotations* $(h_1, ..., h_{T_x})$. Each annotation h_i contains the information about the whole input sentence with a strong focus on the parts surrounding the i^{th} word of the input sequence.
- The proposed model is composed of a BiRNN (Bidirectional RNN), which consists of forward as well as backward RNNs.
- A usual RNN reads an input sequence x from the first symbol x_1 to the last symbol x_{T_x}, where T_x represents the length of the input sequence.

  In the BiRNN, the forward RNN $\overrightarrow{f}$ functions the same way as an ordinary RNN, thus calculating a series of hidden states $(\overrightarrow{h}_1, ..., \overrightarrow{h}_{T_x})$.

  The backward RNN $\overleftarrow{f}$ reads the sequence in reverse order (from x_{T_x} to x_1) calculating a series of hidden states $(\overleftarrow{h}_1, ..., \overleftarrow{h}_{T_x})$.
- The annotation corresponding to the i^{th} word is then defined as $h_i = [\overrightarrow{h}_i ; \overleftarrow{h}_i]^T$. This way, an annotation contains summaries of both preceding as well as following words, with focus on the nearby words due to the tendency of RNNs to better represent recent inputs.

Decoder

- In this model, an alignment model a parametrized as a feedforward neural network which is jointly trained with all the other components of the proposed system, is defined. This model scores how well the inputs around position j and output at position i match.

$$e_{ij} = a(s_{i-1}, h_j)$$

- The weight α_{ij} of each annotation h_j is computed by:

$$\alpha_{ij} = \frac{exp(e_{ij})}{\sum_{k=1}^{T_x} exp(e_{ik})}$$

- In this model, context vectors c_i are computed as a weighted sum of annotations h_i:

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j$$

  Let α_{ij} be a probability that the target word y_i is aligned to, or translated from, a source word x_j. Then, the i^{th} context vector c_i is the expected annotation over all the annotations with probabilities α_{ij}.
  e_{ij} can be seen as the energy associated with probability α_{ij}.

- The conditional probability of the next word in the output sequence is represented as:

$$p(y_i|y_1, ..., y_{i-1}, x) = g(y_{i-1}, s_i, c_i)$$

  Where s_i is the RNN hidden state at i^{th} timestep, computed by:

$$s_i = f(s_{i-1}, y_{i-1}, c_i)$$

  Therefore, α_{ij} or e_{ij} reflects the importance of the annotation h_j with respect to the previous hidden state s_{i-1} in deciding the next state s_i and generating y_i.
  Hence, the probability for each target word y_i is conditioned on a distinct context vector c_i.

Key Points

- Unlike traditional NMT, alignment is not considered to be a latent variable here.
- Instead, the alignment model directly computes a soft alignment, which allows the gradient of the cost function to be backpropagated through. This gradient can be used to train the alignment model as well as the whole translation model jointly.
- Taking a weighted sum of all the annotations can be seen as computing an expected annotation, where the expectation is over possible alignments.

Experiments

1. The approach is evaluated on the task of English-to-French translation on the bilingual, parallel corpora provided by ACL WMT' 14.

Dataset

- WMT '14 contains the following English-French parallel corpora: Europarl (61M words), news commentary (5.5M), UN (421M) and two crawled corpora of 90M and 272.5M words respectively, totaling 850M words.
- Following the procedure described in Cho et al. (2014a), the size of the combined corpus is reduced to have 348M words using the data selection method by Axelrod et al. (2011).

- news-test-2012 and news-test-2013 are concatenated to make a development (validation) set, and evaluate the models on the test set (news-test-2014) from WMT '14, which consists of 3003 sentences not present in the training data.
- After a usual tokenization, a shortlist of 30,000 most frequent words in each language is used to train the models. Any word not included in the shortlist is mapped to a special token ([UNK]). No other special preprocessing, such as lowercasing or stemming, is applied to the data.

2. Two types of models are trained: the first one is an RNN Encoder–Decoder, and the other is the proposed model, referred to as RNNsearch.
3. Each model is trained twice: first with the sentences of length up to 30 words (RNNencdec-30, RNNsearch-30) and then with the sentences of length up to 50 word (RNNencdec-50, RNNsearch-50).
4. The encoder and decoder of the RNNencdec have 1000 hidden units each. The encoder of the RNNsearch consists of forward and backward recurrent neural networks (RNN) each having 1000 hidden units. Its decoder has 1000 hidden units.
5. In both cases, a multilayer network with a single maxout hidden layer is used to compute the conditional probability of each target word.
6. A minibatch SGD algorithm together with Adadelta is used to train each model. Each SGD update direction is computed using a minibatch of 80 sentences. Each model was trained for approximately 5 days.
7. Once a model is trained, a beam search is used to find a translation that approximately maximizes the conditional probability.

Results

Quantitative Results

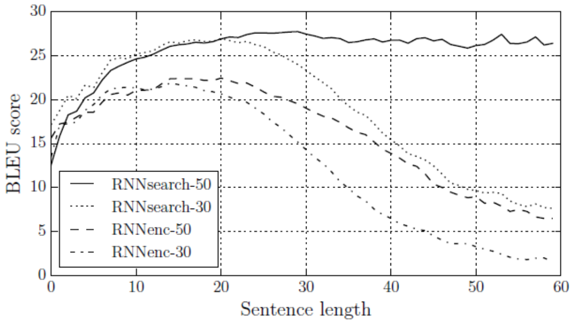


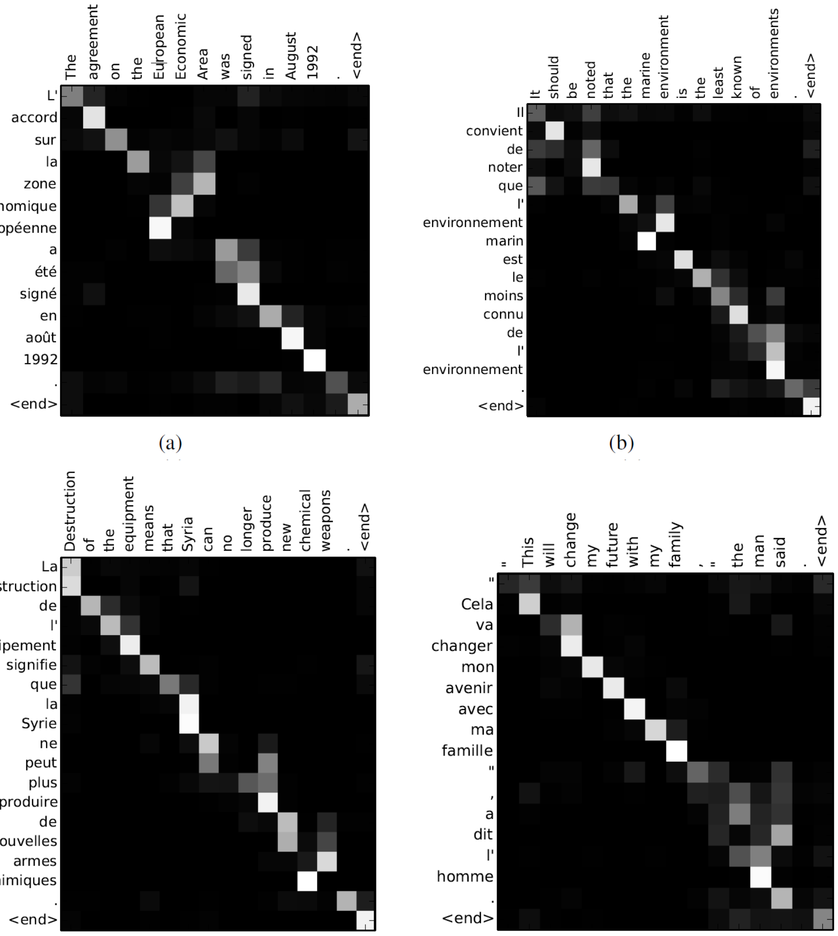
Figure 2: The BLEU scores of the generated translations on the test set with respect to the lengths of the sentences. The results are on the full test set which includes sentences having unknown words to the models.

Model	All	No UNK°
RNNencdec-30	13.93	24.19
RNNsearch-30	21.50	31.44
RNNencdec-50	17.82	26.71
RNNsearch-50	26.75	34.16
RNNsearch-50*	28.45	36.15
Moses	33.30	35.63

Table 1: BLEU scores of the trained models computed on the test set. The second and third columns show respectively the scores on all the sentences and, on the sentences without any unknown word in themselves and in the reference translations. Note that RNNsearch-50* was trained much longer until the performance on the development set stopped improving. (○) We disallowed the models to generate [UNK] tokens when only the sentences having no unknown words were evaluated (last column).

- Moses is a conventional phrase-based translation system.
- Outperforming Moses (when sentences of known words are considered) is a significant achievement as Moses uses a separate monolingual corpus (418M words) in addition to the parallel corpora we used to train the RNNsearch and RNNencdec.

Qualitative Results



(a)

(b)

- The proposed approach provides an intuitive way to inspect the (soft-)alignment between the words in a generated translation and those in a source sentence. This is done by visualizing the annotation weights α_{ij} from as in the figure above. Each row of a matrix in each plot indicates the weights associated with the annotations. From this we see which positions in the source sentence were considered more important when generating the target word.

- It can be seen from the alignments that the alignment of words between English and French is largely monotonic, which is evident from strong weights along the diagonal of each matrix.

- Also, a number of non-trivial, non-monotonic alignments can be observed, which is where the strength of soft alignment can be seen. Adjectives and nouns are typically ordered differently between French and English. From this figure, it can be observed that the model correctly translates a phrase [European Economic Area] into [zone economique europeen]. The RNNsearch was able to correctly align [zone] with [Area], jumping over the two words ([European] and [Economic]), and then looked one word back at a time to complete the whole phrase [zone economique europeenne].

Bibliography

1. Axelrod, A., He, X., and Gao, J. (2011). Domain adaptation via pseudo in-domain data selection. In Proceedings of the ACL Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 355–362. Association for Computational Linguistics.

2. Bastien, F., Lamblin, P., Pascanu, R., Bergstra, J., Goodfellow, I. J., Bergeron, A., Bouchard, N., and Bengio, Y. (2012). Theano: new features and speed improvements. Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop.

3. Bengio, Y., Simard, P., and Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. IEEE Transactions on Neural Networks, 5(2), 157–166.

4. Bengio, Y., Ducharme, R., Vincent, P., and Janvin, C. (2003). A neural probabilistic language model. J. Mach. Learn. Res., 3, 1137–1155.

5. Bergstra, J., Breuleux, O., Bastien, F., Lamblin, P., Pascanu, R., Desjardins, G., Turian, J., Warde-Farley, D., and Bengio, Y. (2010). Theano: a CPU and GPU math expression compiler. In Proceedings of the Python for Scientific Computing Conference (SciPy). Oral Presentation.

6. Boulanger-Lewandowski, N., Bengio, Y., and Vincent, P. (2013). Audio chord recognition with recurrent neural networks. In ISMIR.

7. Cho, K., van Merrienboer, B., Gulcehre, C., Bougares, F., Schwenk, H., and Bengio, Y. (2014a). Learning phrase representations using RNN encoder-decoder for statistical machine translation. In Proceedings of the Empiricial Methods in Natural Language Processing (EMNLP 2014). to appear.

8. Cho, K., van Merri¨enboer, B., Bahdanau, D., and Bengio, Y. (2014b). On the properties of neural machine translation: Encoder–Decoder approaches. In Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation. to appear.

9. Devlin, J., Zbib, R., Huang, Z., Lamar, T., Schwartz, R., and Makhoul, J. (2014). Fast and robust neural network joint models for statistical machine translation. In Association for Computational Linguistics.

10. Forcada, M. L. and N˜ eco, R. P. (1997). Recursive hetero-associative memories for translation. In J. Mira, R. Moreno-D´ıaz, and J. Cabestany, editors, Biological and Artificial Computation: From Neuroscience to Technology, volume 1240 of Lecture Notes in Computer Science, pages 453–462. Springer Berlin Heidelberg.

11. Goodfellow, I., Warde-Farley, D., Mirza, M., Courville, A., and Bengio, Y. (2013). Maxout networks. In Proceedings of The 30th International Conference on Machine Learning, pages 1319–1327.

12. Graves, A. (2012). Sequence transduction with recurrent neural networks. In Proceedings of the 29th International Conference on Machine Learning (ICML 2012).

13. Graves, A. (2013). Generating sequences with recurrent neural networks. arXiv:1308.0850 [cs.NE].

14. Graves, A., Jaitly, N., and Mohamed, A.-R. (2013). Hybrid speech recognition with deep bidirectional LSTM. In Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on, pages 273–278.

15. Hermann, K. and Blunsom, P. (2014). Multilingual distributed representations without word alignment. In Proceedings of the Second International Conference on Learning Representations (ICLR 2014).

16. Hochreiter, S. (1991). Untersuchungen zu dynamischen neuronalen Netzen. Diploma thesis, Institut f¨ur Informatik, Lehrstuhl Prof. Brauer, Technische Universit¨at M¨unchen. Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735–1780.

17. Kalchbrenner, N. and Blunsom, P. (2013). Recurrent continuous translation models. In Proceedings of the ACL Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1700–1709. Association for Computational Linguistics.

18. Koehn, P. (2010). Statistical Machine Translation. Cambridge University Press, New York, NY, USA.

19. Koehn, P., Och, F. J., and Marcu, D. (2003). Statistical phrase-based translation. In Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1, NAACL '03, pages 48–54, Stroudsburg, PA, USA. Association for Computational Linguistics.

20. Pascanu, R., Mikolov, T., and Bengio, Y. (2013a). On the difficulty of training recurrent neural networks. In ICML'2013.

21. Pascanu, R., Mikolov, T., and Bengio, Y. (2013b). On the difficulty of training recurrent neural networks. In Proceedings of the 30th International Conference on Machine Learning (ICML 2013).

22. Pascanu, R., Gulcehre, C., Cho, K., and Bengio, Y. (2014). How to construct deep recurrent neural networks. In Proceedings of the Second International Conference on Learning Representations (ICLR 2014).

23. Pouget-Abadie, J., Bahdanau, D., van Merri¨enboer, B., Cho, K., and Bengio, Y. (2014). Overcoming the curse of sentence length for neural machine translation using automatic segmentation. In Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation. to appear.

24. Schuster, M. and Paliwal, K. K. (1997). Bidirectional recurrent neural networks. Signal Processing, IEEE Transactions on, 45(11), 2673–2681.

25. Schwenk, H. (2012). Continuous space translation models for phrase-based statistical machine translation. In M. Kay and C. Boitet, editors, Proceedings of the 24th International Conference on Computational Linguistics (COLIN), pages 1071–1080. Indian Institute of Technology Bombay.

26. Schwenk, H., Dchelotte, D., and Gauvain, J.-L. (2006). Continuous space language models for statistical machine translation. In Proceedings of the COLING/ACL on Main conference poster sessions, pages 723–730. Association for Computational Linguistics.

27. Sutskever, I., Vinyals, O., and Le, Q. (2014). Sequence to sequence learning with neural networks. In Advances in Neural Information Processing Systems (NIPS 2014). Zeiler, M. D. (2012). ADADELTA: An adaptive learning rate method. arXiv:1212.5701 [cs.LG].