NLP Paper Review
Neural Machine Translation by Jointly Learning to Align and Translate

Aditeya Baral

February 2021

1 Authors

Dzmitry Bahdanau, KyungHyun Cho, Yoshua Bengio

2 Abstract

1. Neural Machine Translation ordinarily uses an encoder-decoder architecture – the encoder encodes the source sentence into a fixed length vector and the decoder decodes the vector into a target sentence
2. Their model automatically soft-searches for parts of a sentence relevant to predicting a word
3. The soft alignments found by model were on par with human intuition
4. Their model was able to produce state-of-the-art results

3 Introduction

1. NMT models use encoder-decoder architecture – an encoder and decoder for each language to be translated
2. The whole encoder–decoder system, which consists of the encoder and the decoder for a language pair, is jointly trained to maximize the probability of a correct translation given a source sentence.
3. An issue with this encoder–decoder approach is that a neural network needs to be able to compress all the necessary information of a source sentence into a fixed-length vector – does not work well with long sequences (performance of encoder-decoder decreases rapidly with length of input sequence)

4. Each time the proposed model generates a word in a translation, it (soft-)searches for a set of positions in a source sentence where the most relevant information is concentrated

5. The model then predicts a target word based on the context vectors associated with these source positions and all the previous generated target words

6. 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.

4 Previous Work

1. RNN based encoder-decoder architectures were used

2. Hybrid models such as RNN and de-convolutional neural networks have also been tested

5 Approach

1. Encoder – The encoder consisted of a bidirectional RNN. The annotation for each word is formed by concatenating the forward and hidden states h(i), and this annotation will contain information from both forward and backward context – but will be more focused around x(i)

2. Decoder –

   (a) A context vector is calculated as a weighted sum of the sequence of annotations h to which the encoder maps the input sequence . Each annotation contains information about the whole input sequence with a strong focus on the parts surrounding the i-th word of the input sequence.

   (b) We can understand the approach of taking a weighted sum of all the annotations as computing an expected annotation, where the expectation is over possible alignments. Let alpha 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 alpha(i, j) .

   (c) The weight of each annotation is calculated using a softmax of the alignment model

   (d) The alignment model scores how well inputs and outputs about positions match. This score depends on the hidden state of the decoder

RNN just before the output position and the annotations produced by the encoder RNN

(e) The alignment model itself is jointly trained with a feed forward neural network, and directly computes a soft alignment

(f) alpha (i, j) refers to the importance, or attention paid to h(j) with respect to the previous decoder state s(i-1) to decide the next decoder state s(i) and output y(i)

6 Data

1. The Dataset used was a English-French translation dataset, provided by ACL WMT '14
2. It consists of Europarl, news commentary and other types of sources
3. The dataset size was reduced to 348M words, and the validation set contained 3003 sentences
4. During tokenization, only the top 30,000 words were used and the remaining words were mapped to UNK

7 Novelties

1. It does not use a fixed-length context vector built by the encoder to perform translation
2. It used soft-alignment to find positions of input words where most relevant information is concentrated (words crucial to predicting the output word) along with previously generated words
3. It encodes the input sentence into a sequence of vectors and chooses these vectors while performing the translation
4. This frees a neural translation model from having to squash all the information of a source sentence, regardless of its length, into a fixed-length vector – and hence coped better with longer sentences

8 Evaluation

1. Comparisons were made using the BLEU score, and their model outperformed the conventional encoder-decoder
2. It achieved performance as high as the phrase based translation system - Moses

3. Their model was robust to increase in sequence length, showing no performance decrease on sentences with more than 50 words – it still provided more accurate translations

4. The soft alignments (alpha) were visualised, and observed. Positions in the source sentence critical to generating the target word were observed and found to be intuitively correct.

9 Summary

The model replaces the standard encoder-decoder architecture with a new model that does not encode the entire input sequence into a context vector. Instead, an alignment model is made using a feed forward neural network, which learns the relation between the hidden states in the encoder and decoder. Through this, weights for each encoder hidden state annotation is calculated – this annotation stores information about how relevant, crucial or how much attention to pay to a certain word in the input while predicting a word in the output. From the annotations, a context vector is finally prepared which is used by the decoder state to predict a word in the output sequence. It thus learns to align words in the input sequence, based on relevance to words being predicted in the output sequence and uses this as well as previously generated words to predict the next word (unlike the regular model which directly used a context vector constructed from the entire input sequence). The model outperforms the regular encoder-decoder model, as well as obtains performance on the same level as phrase based systems. Additionally, the attention weights obtained after aligning were found to be intuitively accurate as well.