Sentence Simplification

**Abstract:**

Sentence simplification aims to simplify the content and structure of complex sentences, and thus make them easier to interpret for human readers, and easier to process for downstream NLP applications. In this paper, we adapt an architecture of Encoder-Decoder model presented by Facebook AI in []. Facebook's model was originally developed for Neural Machine Translation, however, we modified it for the sentence simplification task.

**Introduction:**

The goal of sentence simplification is to convert complex sentences into simpler ones so that they are more understandable and accessible, while still keeping their original information content and meaning. Sentence simplification has a number of practical applications: it is useful for bilingual education and other language-learning contexts. It can help patients with linguistic and cognitive disabilities (Carroll et al., 1999). Sentence simplification can also be used to improve performance in other NLP tasks (Niklaus et al.(2017). [Chandrasekar et al., 1996; Knight and Marcu, 2000; Beigman Klebanov et al., 2004]

**Related Work:**

In previous studies, researchers of sentence-level simplification mostly address the simplification task as a machine translation problem. Specia et al. (2010) use statistical machine translation approach implemented in Moses toolkit (Koehn et al., 2007) to translate the original sentences to the simplified ones. Wang et al. (2016) were the first to suggest using a NMT model for text simplification. They used a LSTM encoder - decoder seq2seq model, but due to the lack of an adequate dataset they used a number-based sequences instead of natural language data. Coster et al.(2011) introduced a new dataset of aligned sentence pairs taken from Wikipedia and Simple English Wikipedia, the dataset is widely used in many sentence simplification researches. Zhang et al.(2017) suggested a constrained seq2seq neural model for sentence simplification, their model combines world level and sentence level simplifications and yields better results than various baselines. Meng et al.(2015) proposed using a convolutional neural network to encode the source language for NMT. Our work is based on the model that was presented by Gehring et al.(2017) for NMT, which uses two convolutional neural networks as an encoder, and an attention based recurrent neural network as the decoder.(Flavio?)

**Datasets**

Two main datasets were considered for our model:

Simple English Wikipedia (Coster et al.(2011)): A sentence aligned dataset taken from parallel articles in English Wikipedia and Simple English Wikipedia. This dataset contains 167K pairs of sentences and is one of the largest datasets used for sentence simplification. While examining this dataset we noticed a few problems – Many sentences contain special characters, URLs, gibberish, excess use of punctuation and more. <example>. Such anomalies can interfere the training procedure and cause unreliable results.

Newsela (Xu et al.(2015)):

A simplification corpus of news articles, re-written by professional editors to meet the readability standards for children at multiple grade levels. Each sentence in the corpus is rewritten in up to 6 different level of complexity. The creators of this dataset mapped all the problems that exist in the Simple Wikipedia corpus and addressed them in their research. The Newsela dataset contains 141K pairs of aligned sentences.

Our model supports both datasets but because of the problems we mentioned above we used the Newsela corpus for training and evaluation.

Data preparation

To use the data we needed some pre-processing. Two aligned lists of sentences were constructed from the raw data. From each list a vocabulary which maps each word to a unique integer ID was created. Using the mentioned vocabularies, every sentence was converted to a list of word IDs. Each tokenized sentence is fed later as input to our model, which uses GloVe embeddings (Pennington et al.(2014)) to represent each word in lower dimensional space R^100.

**Our approach:**

We chose to adapt a NMT model to the sentence simplification task. Most of the seq2seq neural models we encountered were based on RNN encoder – decoder, however we decided to encode the source sentences with a Convolutional Neural Network instead.

Facebook AI (Gehring et al.(2017)) used a similar approach for NMT. Di Palo et al.(2019) tried it too for the sentence classification. But as far as we know, we are the first to try this architecture for sentence simplification.

CNNs computation, contrary to RNNs, can be parallelized, optimization is easier since the number of non-linearities is fixed and independent of the input length and last because they outperform the LSTM accuracy in [Wu et al., (2016).](https://arxiv.org/abs/1609.08144)

Encoder Architecture

As suggested by Gehring et al.(2017) The encoder consists of two stacked convolutional networks: CNN-a’s output zj used for creating the attention matrix A that is used at decoding time. Simultaneously, CNN-c’s output z’j is used to produce the conditional input ci by a simple dot product between the attention vector ai with it.<formula>. The CNNs do not contain pooling layers which are commonly used for down-sampling, i.e., the full source sequence length will be retained after the networks has been applied. <figure>.

One of the challenges of using CNNs encoders is the loss of word ordering. In order to solve it, Gehring et al.(2017) proposes to use position embeddings in addition to the pretrained word embeddings <formula> <table>

Decoder Architecture

* hi represents the hidden state/output of the LSTM.
* ci is the input context to the LSTM
* gi is the embedding of the previous output of the LSTM. This gets concatenated with ci as input to the LSTM

We use LSTMs (Hochreiter and Schmidhuber, 1997) for all decoder networks whose state si comprises of a cell vector and a hidden vector hi which is output by the LSTM at each time step. We input ci into the LSTM by conc`atenating it to gi.

The translation model computes a distribution over the V possible target words yi+1 by trans- forming the LSTM output hi via a linear layer with weights Wo and bias bo. <formula><figure>

we transform the decoder hidden state hi by a linear layer with weights Wd and bd to match the size of the embedding of the previous target word gi and then sum the two representations to yield di <formula>. Conditional input ci is a weighted sum of attention scores ai ∈ Rm and encoder outputs z <formula>. The attention scores ai are determined by a dot product between hi with each zj, followed by a softmax over the source sequence:<formula>

<figure from poster>

Experimental setup

Control – test simple RNN enc dec, describe results

Test our model –

Overfit our model for sanity check

Run on full dataset (describe parameters used)

Optimization

Parameters tuning (cite article)

Teacher forcing

Custom loss?

Weighted sum instead of argmax (cite jonathan)

Future work

Loss?

Beam search

Optimize code for parallelism (multiple GPUs etc)

More epochs maybe on faster system

<https://arxiv.org/pdf/1804.07445.pdf> SentenceSimpliﬁcationwithMemory-AugmentedNeuralNetworks

<https://arxiv.org/pdf/1904.02767.pdf> Complexity-WeightedLossandDiverseReranking forSentenceSimpliﬁcation

<https://arxiv.org/pdf/1906.05483.pdf> Enriching Neural Models with Targeted Features for Dementia Detection

<https://arxiv.org/pdf/1411.4389.pdf> Long-term Recurrent Convolutional Networks for Visual Recognition and Description

<https://arxiv.org/pdf/1611.02344.pdf> A Convolutional Encoder Model for Neural Machine Translation

<https://arxiv.org/pdf/1503.01838.pdf> Encoding Source Language with Convolutional Neural Network for Machine Translation

<https://arxiv.org/pdf/1805.05557.pdf> Simplifying Sentences with Sequence to Sequence Models

<https://arxiv.org/pdf/1510.03820.pdf> A Sensitivity Analysis of (and Practitioners’ Guide to) Convolutional Neural Networks for Sentence Classification

<https://arxiv.org/pdf/1704.02312.pdf> A Constrained Sequence-to-Sequence Neural Model for Sentence Simplification

<https://arxiv.org/pdf/1703.10931.pdf> Sentence Simplification with Deep Reinforcement Learning

<https://arxiv.org/pdf/1703.09013.pdf> A Sentence Simplification System for Improving Relation Extraction

<https://arxiv.org/pdf/1609.03663.pdf> An Experimental Study of LSTM Encoder-Decoder Model for Text Simplification

<https://arxiv.org/pdf/1507.08452.pdf> Unsupervised Sentence Simplification Using Deep Semantics

[<https://arxiv.org/pdf/1703.09013.pdf>] A Sentence Simplification System for Improving Relation Extraction. Niklaus et al.(2017)

Use cases

* Common models
* Our approach
* Why we chose that
* Our contribution