Variable length word encodings for neural translation models

Jiameng Gao
University of Cambridge

UNIVERSITY OF CAMBRIDGE

Overview

We look at methods of word decompositions through the perspective of vocabulary compression. A vast proportion of words or tokens in most natural languages occur very infrequently, meaning that these words would be poorly trained for in a neural network model for any given corpus of finite size. This is known as the *rare word problem*.

Further, the softmax function (Eq. 1) used on the output layers of neural networks it greatly slows down neural network training and evaluation due to its arithmetic complexity, especially as the number of outputs (i.e. the vocabulary size) $|\mathcal{I}|$ increases.

$$\alpha_i = \frac{\exp \langle \mathbf{w}_{\mathbf{l}-\mathbf{1},\mathbf{i}}, \mathbf{h}_\mathbf{l} \rangle}{\sum_{i' \in \mathcal{I}} \exp \langle \mathbf{w}_{\mathbf{l}-\mathbf{1},\mathbf{i}'}, \mathbf{h}_\mathbf{l} \rangle} \tag{1}$$

A common method to counteract for this is to limit the source and target vocabulary to a fixed number. However, doing so only restricts the ability of the translation system to produce robust outputs of a desired quality, as outputs may contain `UNK` symbols in place of words important to the whole sentence. This is the *unknown word problem*.

Neural machine translation

The first uses of neural networks in machine translation came in the form of neural network based language models. Neural networks language models can often out-perform Kneser-Ney N-gram based language models and are used for rescoring or decoding purposes.

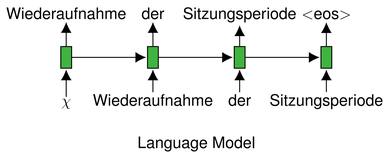

Figure 1: A RNN Language Model

Nonetheless, recent developments in neural network based methods have allowed for end-to-end machine translation using neural networks similar to an autoencoder. Sutskever et al. (2014) first proposed a neural network encoder-decoder structure, composed of an encoder which produces a vector representation of the source sentence, and a decoder that outputs words in the target language.

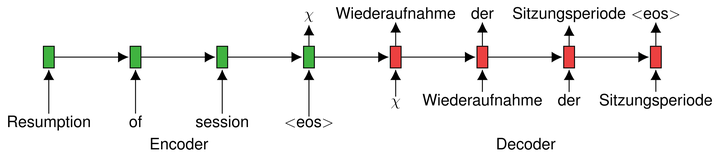

Figure 2: The encoder-decoder translation system from Sutskever et al. (2014)

Implementation

- TensorFlow - scalable with CUDA support
- LSTM RNN as architecture for language models and end-to-end translation
- Rescore N-best lists with RNN LM
- Implement lattice-based rescoring through OpenFST

Huffman encoding

Chitnis and Denero (2015) employed the equivalent of Huffman coding to the rare words, thereby decomposing and replacing them with a sequence of commonly shared symbols (or "words"), while preserving a list of common words.

For example, for the sentence *the cat hit the dog* and a coding alphabet size of 3 (`s_0`, `s_1`, `s_2`), the decomposition is as follows:

Figure 3: Example of the Huffman coding scheme

Byte-pair encoding

Byte pair encoding is a compression method that attempts to replace the most frequent sequences with symbols. This technique is applied in Sennrich et al. 2015 construct subword units by looking at the most frequent character pairs - discovering subword units by building up pairs from the character level.

As an example, if the words *low*, *lower* and *slower* were in the corpus, occurring once, then the merge operations are the sequence below in order of the most frequent pairs, also showing the state of the vocabulary:

```
(l o w </w>) (l o w e r </w>) (s l o w e r </w>)  l o --> lo
(lo w </w>) (lo w e r </w>) (s lo w e r </w>)     lo w --> low
(low </w>) (low e r </w>) (s low e r </w>)        e r --> er
(low </w>) (low er </w>) (s low er </w>)          er </w> --> er</w>
(low </w>) (low er</w>) (s low er</w>)            low er</w> --> lower</w>
(low </w>) (lower</w>) (s lower</w>)
```

Importance sampling

Jean et al. (2015) and Ji et al. (2015) suggest importance sampling methods to decorrelate the computation expense with the size of the vocabulary. Both of the methods proposed to divide the target vocabulary into subsets $V_i \subset V$, such that during training we only perform gradient updates with respect to the correct word and the subset of the vocabulary V_i.

Copy from source text

Luong et al. (2014) attempted to address the problem by marking the location of these OOV words and its alignment to the source language. They then translate the words from the source language using a more general purpose translation dictionary, or copy aligned words from the source input if no translation is found.

Character level decomposition

At the same time, Luong et al. (2016) work around the rare and unknown word problem by creating a "hybrid" NMT system using a standard backbone with a character-to-word encoder and a separate word-to-character decoder.

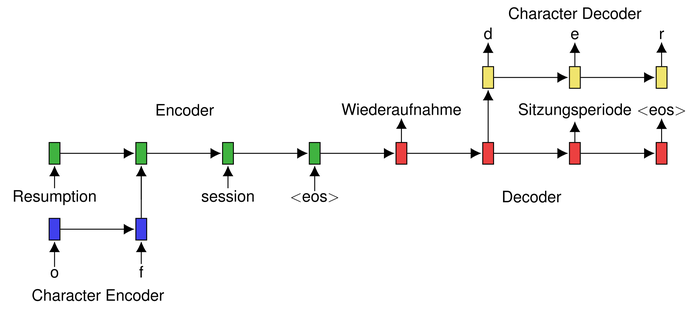

Figure 4: The hybrid neural translation system by Luong et al. (2016)

Meanwhile, Chung et al. (2016) approaches the problem from a neural network design perspective; they devise a "bi-scale recurrent neural network" that uses two Gated Recurrent Units that capture dependencies at a fast and a slow timescale.

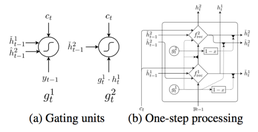

Figure 5: Bi-scale RNN. Taken from Chung et al. (2016)

Linguistic based decomposition

There are readily available morphemic decomposition tools, but further, perhaps we could utilise phonological decomposition, since phones are better units to form morphemes than characters.

```
BACK --> B AE K
CALL --> K AO L
NO --> N OW
NUMBER --> N AH M B ER
```