Using Neural Networks for Machine Translation

Abstract

Machine translation using recurrent neural networks (RNN) has replaced statistical phrase based models for several production translation systems. For this paper I created a RNN using word embeddings, three stacked GRU cells, an attention mechanism, and sampled softmax to translate English into Portuguese. The efficiency of translation was determined by comparing the RNN model with the baseline model, which is a statistics based model built using Moses. While the RNN model I trained was able to translate some words, there was an issue with outputting a large number of unknown word tokens. For future work this model could be improved by word stemming, increasing vocabulary size, or training a deeper model.

Introduction

The first attempts at machine translation started using bilingual dictionaries and applying fixed, hand curated rules for grammar and word reordering. These early attempts could not consistently produce natural sounding translations across an entire language because there are simply too many rules and exceptions to enumerate. Language is incredibly complex, and translation even more so. In order for humans to correctly translate something from one language to another fluently requires us to not just know the translation of each word in a sentence, but in most cases to actually understand meaning and context.

There is innate ambiguity in most languages and it was believed that only fundamental understanding of words and their meanings could ensure correct translations. To combat the issue of language ambiguity, systems were developed to translate languages first into some unambiguous intermediary, and then into the target language. While this improved results, there are so many language and each language is so vast that it is simply not feasible to write rules for disambiguation and translation by hand. This gave rise to the field of Statistical Machine Translation (SMT). By analyzing large parallel corpuses, machines can gather statistics on what translations are more likely than others.

While SMT was the state of the art until very recently, neural network models for machine translation have become very popular. While statistical phrase based translation works by translating groups of words in a sentence and then ordering those translated groups properly to form an output, neural networks allow for a single end-to-end model to process the entire sentence at once. Furthermore, many production translation systems are transitioning over to these newer models citing improved translation accuracy and quality (Wu, Y. et al. 2016).

The goal is to create a system to translate text from English to Portuguese using a sequence to sequence recurrent neural network model. To create this model, I have adapted a basic sequence to sequence model (Cho et al., 2014, Sutskever et al., 2014) and incorporated an attention mechanism (Bahdanau et al., 2014). The model consists of three recurrent neural

networks (RNNs), one for encoding the English sentence, one for decoding into Portuguese, and an attention mechanism to allow the decoder to selectively use different parts of the encoder's output. The encoder and decoder will both be composed of 3 stacked GRU cells. To train my model, I used the English-Portuguese dataset from Europarl: A Parallel Corpus for Statistical Machine Translation (Koehn, P., 2005). The performance of the model is evaluated against a baseline SMT system on sample sentences by a native speaker.

Methods

Data and Vocabulary

The dataset used to train the model was the English-Portuguese parallel corpus from Europarl (Kohen, 2005). This dataset consists of 2M parallel sentences, 50M English words, and 50M Portuguese words. An exploratory analysis of the data was performed to determine an optimal method of tokenization and vocabulary size. For a strict tokenizer we split words using the NLTK's (Natural Language ToolKit) word_tokenizer and used this output directly. For a more flexible tokenizer we lowercased our inputs and ignored numbers and punctuation. For either tokenizer the Portuguese vocabulary contained more unknown words at a fixed vocabulary size. As expected the more flexible tokenizer resulted in a smaller percentage of unknown words, allowing us to use a smaller vocabulary size to speed up training (Figure 1). Of course the cost of this is the output lacks capitalization and punctuation, but since there was limited time for training this seemed like a reasonable tradeoff.

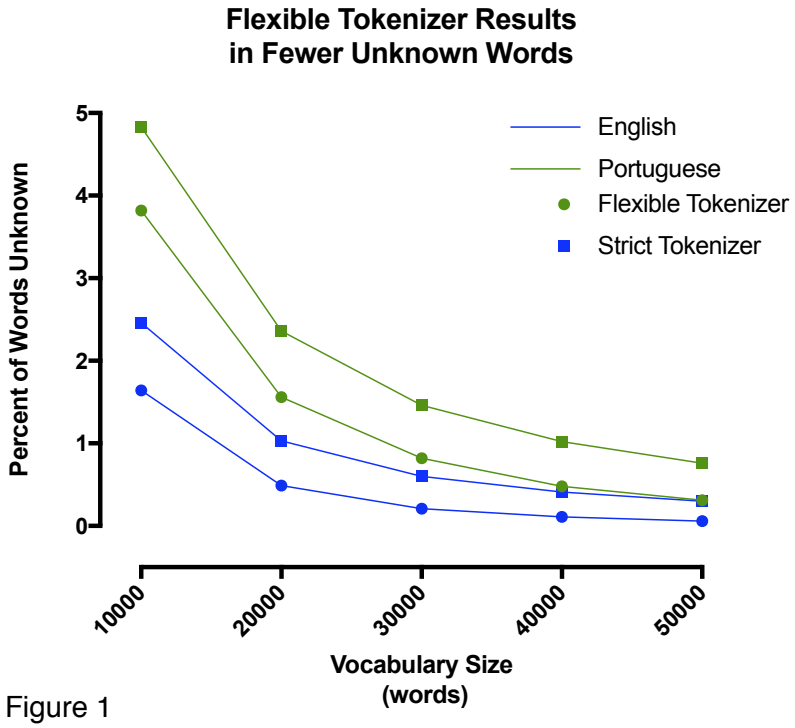


Figure 1

Mapping

Now that the input is tokenized into sequences of words, the next step is to convert those sequences into vectors to train the model. The vocabulary consists of the 20,000 most common English and 40,000 most common Portuguese words along with 4 special symbols: the unknown word symbol (_UNK), the padding symbol (_PAD), the go symbol (_GO), and the end of sentence symbol(_EOS). The unknown word symbol is substituted for words that are not in the vocabulary. The vocabulary sizes were chosen such that this would be < 1% of all tokens. The pad symbol is used to make inputs and outputs fit inside of our fixed length buckets. The go symbol is inserted at the start of the Portuguese decoder inputs and the end of sentence symbol at the end of the decoder inputs.

Bucketing

The training sentences vary in length from a single word all the way up to very long run on sentences. Tensorflow does not currently support variable length sequences in RNNs, so this is a problem. In order to deal with these variable length inputs and outputs the data could be padded to all be the same length as the longest input and output, but this would cause there to be a lot of pad symbols in our training data and would probably mess with the output. We could also create a model for each possible length of input and output, but that would create a very large number of modules and require a much larger amount of training data. The way the Tensorflow framework recommends dealing with this problem is through bucketing. Bucketing is where a model is created for each of a small number of fixed size buckets, and the input and output pairs are put into the smallest bucket they fit into, and then padded with the pad symbol so they are the correct size. The model used the four bucket sizes (Table 1).

Input	Output
5	10
10	15
20	25
40	50

Table 1

Embedding

The embedding of the input vector into a smaller dimensional space can really be thought of as the first layer of the network. Embedding is a very useful tool across a wide variety of NLP tasks. The trained embedding of the English words contains semantic information, greatly improving the ability of the model to understand them. Tensorboard produces a nice visualization of the trained embedding using principle component analysis (Figure 2).

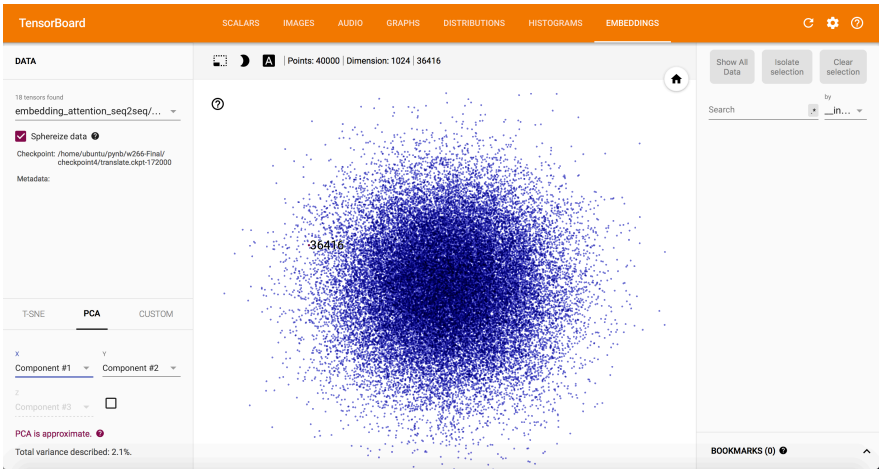
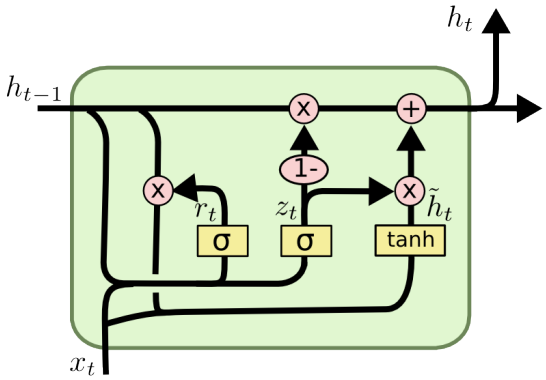


Figure 2

Encoding

Now the model encodes the embedded vectors into a hidden state vector that summarizes the meaning of the input sentence. The process of creating that vector is simply feeding the input into our recurrent neural network encoding model. The model is composed of three stacked layers. In each layer is a gated recurrent unit (GRU) cell which has been shown to perform as well as the more complicated long short term memory (LSTM) cell (Chung, et al. 2014). A GRU is a recurrent network cell that takes in a vector as input, as well as the output of the previous time step, which we will call the hidden state or h. At t=0 the hidden state is just the 0-vector. From the inputs we compute two vectors called the update gate and the reset gate. The update gate is used to determine how much of the output should be the hidden state and how much should be a linear function of our input combined with the reset gate times the hidden state (see Figure 3).



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

Figure 3 (http://colah.github.io/posts/2015-08-Understanding-LSTMs/)

Decoder with Attention

The decoder is a very similar model to our encoder, three stacked GRU cells. It is provided the go symbol, the target output sentence, the end of sentence symbol, and finally padding if necessary as input. During training the decoder is provided with the correct input at the previous time step even if that is not the output it produced (Bengio et al., 2015). It also gets another important input, the hidden states of the encoder at various time steps. These inputs are fed to our decoder through an attention mechanism (Bahdanau et al., 2014). This mechanism has trained weights that allow the decoder to focus on specific parts of the input during decoding, significantly improving performance on longer inputs.

Development and Training Hardware

The development of the model was completed locally using a notebook in PyCharm for Python and Tensorflow development and RStudio for exploratory data analysis. For the first attempt to train the network, I used my local machine. However, the training process using my CPU was slow. After running for several days only a small amount of progress was made, to address this issue I sucked it up and in the name of science and learning provisioned one of the shiny new P2 instances (https://aws.amazon.com/ec2/instance-types/p2/) from AWS. This also had the benefit of letting me use a nice pre-configured AMI, for the record I went with the Bitfusion Ubuntu 14 TensorFlow (https://aws.amazon.com/marketplace/pp/B01EYKBEQ0) image.

Running on the P2.xlarge with its 61 gigs of memory and 2,496 parallel processing cores I saw a huge improvement in training time.

Results

To create a baseline to compare our model against I used Moses, a popular statistical machine translation framework. I followed the baseline tutorial found here http://www.statmt.org/moses/?n=Moses.Baseline. Unfortunately, the baseline model I built translates from Portuguese to English and not the other way around, but we can still compare the general translation quality of the two systems.

We trained our RNN model for over 200,000 global steps, our final learning rate had decayed to 0.0712, our model had a perplexity of 2.98. On the development set the smallest bucket (5,10) had perplexity 1.65, the second bucket (10,15) had perplexity 2.80, the third bucket (20,25) had perplexity 4.83, and the largest bucket (40,50) had perplexity 5.92. Despite this, our model preformed fairly poorly on our randomly selected test sentences (see Appendix 1), returning a large number of unknown word tokens.

A native Portuguese speaker rated the translations of our baseline model and our neural network model on a scale of 1-5 with 5 being the best. The RNN model scored a total of 25 for an average score of 3.2, while the Moses model scored a total of 37 for an average of 4.1. There was one example sentence the RNN scored higher on, one example sentence the two models got the same score on, and for all other examples the baseline model had a higher score.

Future Work

Our model does very poorly on rare words and produces too many unknown tokens. To address this we could try to implement stemming (Wu, et al. 2016). Another way to address this is to use more training data and to increase the vocabulary size. I think it would be interesting to try a character level output model using the 256 UTF-8 codepoints. Right now we have a large output space and we use sampled softmax to deal with that, but even with accents, capital letters, numbers, and punctuation you still have several orders of magnitude less possible outputs which would make training easier. Of course now the problem you are trying to solve is harder so its a tradeoff, but I believe it is worth exploring.

We are simultaneously learning embeddings with our model, but embeddings don't have to be for our particular task of translation. I would argue that embeddings should be trained separately because there are orders of magnitude more data for a single language on which to train embeddings then there are data contained in parallel corpora. Another thing that would be really interesting to try is adding additional language pairs to the model by way of language tokens at the beginning of output (Johnson, et al. 2016).

A finial potential direction for future study is instead of going deeper with our network, i.e. adding more than 3 layers (Google's production system uses 8 with a bunch of connections to deal with vanishing gradients), we could go wide. What I mean by that is train more models that deal directly with the input in order to produce outputs that could be useful to the decoder. For example something like a part of speech tagger or a named entity recognition model. The idea behind having a very deep network is these things are a part of the network, but the benefit of

training separate models is similar to the benefit of training a separate encoder, we can get more data to train them on.

Conclusion

In conclusion Neural Machine Translation represents a generational leap forward in translation quality from previous state of the art Statistical Machine Translation models. We explored creating a Recurrent Neural Network translation model from English to Portuguese. We created vocabularies, mapped our parallel corpus onto these vocabularies, and separated these mapped files into different length buckets. Then we trained a model that embedded inputs into a smaller dimension, encoded using 3 layers of GRU's, decoded with attention, and used sampled softmax to generate output. Evaluating our model we see that we were unable to improve on the baseline statistical machine translation approach even for training data, but we still learned a lot.

References

1. Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, **Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation**, 2016
2. Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, Yoshua Bengio, **Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation**, 2014
3. Johnson, M., Schuster, M., Le, Q. V., Krikun, M., Wu, Y., Chen, Z., et al. **Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation**, 2016
4. Ilya Sutskever, Oriol Vinyals, Quoc V. Le, **Sequence to Sequence Learning with Neural Networks**, 2014
5. Bengio, S., Vinyals, O., Jaitly, N., & Shazeer, N. **Scheduled Sampling for Sequence Prediction with Recurrent Neural Networks**, 2015
6. Dzmitry Bahdanau, KyungHyun Cho, Yoshua Bengio, **Neural Machine Translation by Jointly Learning to Align and Translate**, 2014
7. Sebastien Jean, Kyunghyun Cho, Roland Memisevic, Yoshua Bengio, **On Using Very Large Target Vocabulary for Neural Machine Translation**, 2014
8. Philipp Koehn, **Europarl: A Parallel Corpus for Statistical Machine Translation**, 2005
9. Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, Evan Herbst, **Moses: Open Source Toolkit for Statistical Machine Translation**, 2007
10. Vinyals, O., Kaiser, L., Koo, T., Petrov, S., Sutskever, I., & Hinton, G. **Grammar as a Foreign Language**, 2014
11. Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. **Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling**, 2014

Appendix 1

English	Competition between the regions will certainly strengthen rather than weaken the European Union.
Portuguese	A concorrência entre regiões vai seguramente fortalecer a União Europeia e não enfraquecê-la.
RNN - 2	a _UNK entre as _UNK _UNK seguramente _UNK e _UNK enfraquecer a _UNK europeia
Moses - 4	the competition between regions will certainly strengthen the European Union and not enfraquecê-la.

English	It was this which inspired us to propose the same thing with regard to state aid.
Portuguese	Foi esse quadro que nos inspirou a propor a adopção do mesmo sistema em relação aos auxílios estatais.
RNN - 4	foi isso que nos deu a ideia de que se _UNK a mesma coisa relativamente aos _UNK estatais
Moses - 5	it was this which inspired us to propose the adoption of the same system in relation to the state aid

English	During the course of our work on this report it became clear that there are persistent problems in the spending areas under the control of our budget.
Portuguese	No decurso do nosso trabalho sobre este relatório, tornouse evidente que se verificarem problemas persistentes a nível da utilização das rubricas orçamentais que eram controladas pelo nosso orçamento.
RNN - 3	durante o nosso trabalho sobre este _UNK claro que se verifica problemas nas _UNK do controlo do controlo do nosso _UNK
Moses - 3	In the course of our work on this relatório, tornouse clear that there are persistent problems in the use of budget lines which were monitored by our together

English	All asylum-seekers must be entitled to a fair hearing and an appeal with suspensory effect.
Portuguese	Qualquer requerente de asilo deve beneficiar do direito a uma audição justa e a um recurso suspensivo.
RNN - 3	_UNK que ter direito a uma _UNK justa e um apelo _UNK _UNK
Moses - 4	every asylum seeker should be entitled to a fair hearing and a resource suspensivo.

English	But I want to add a word of caution.
Portuguese	Mas desejo acrescentar uma palavra de prudência.
RNN - 3	mas quero acrescentar uma _UNK de _UNK
Moses - 4	but I would like to add a word of prudência.

English	All of this must nevertheless be done whilst respecting a broad framework - that of the overall acceptability of any solutions we may devise to achieve these three aims.
Portuguese	Tudo isto respeitando no entanto, um quadro geral que é o quadro geral da aceitabilidade global de todas as soluções que possamos encontrar para levar a cabo estes mesmos três desideratos.
RNN - 3	tudo isto _UNK _UNK no entanto em que se mantenha um quadro global da _UNK global de quaisquer _UNK que possamos _UNK para atingir esses objectivos _UNK
Moses - 4	everything that while respecting the entanto, a general framework that is the overall acceptability of all the solutions that we can devise to achieve these three desideratos.

English	We are looking for solutions.
Portuguese	Estamos à procura de soluções.
RNN - 2	procuramos _UNK
Moses - 4	we are looking for soluções.

English	Madam President, on a point of order.
Portuguese	Senhora Presidente, intervenho para um ponto de ordem.
RNN - 5	senhora presidente relativamente a um ponto de ordem
Moses - 4	Madam Presidente, on a point of ordem.

English	I declare the session of the European Parliament adjourned.
Portuguese	Dou por interrompida a sessão do Parlamento Europeu.
RNN - 4	dou por interrompida a _UNK do parlamento europeu
Moses - 5	I declare adjourned the session of Parliament .