CS 533 Project: Neural Machine Translation

Vatsal Parikh, NetID: vp406

I. Introduction

The project aims to build a small working model of neural machine translator using TensorFlow and Python that performs translation from English to German. The objective here is to use state of the art approach to machine translation by employing recurrent neural networks (RNN). The challenging part of the project was to first understand various deep learning models and then apply them to the problem at hand to get concrete and desirable results.

Neural Machine Translation (NMT) has the potential to truly revolutionize the automated translation industry in the coming years. With extensive research and collaboration taking place in NMT and large technology companies already incorporating NMT in their products, NMT might soon become the de facto approach to machine translation.

II. The Problem / Prior Literature

Machine translation has been one of the oldest problems in Artificial Intelligence. Up until 2014 Statistical Machine Translation (SMT) was the standard in machine translation. However, RNN based approaches began to gain traction. At the shared task for machine translation organized by the Conference on Machine Translation (WMT), only one pure neural machine translation system was submitted in 2015. In 2017, almost all submissions were neural machine translation systems.

SMT and NMT both replace words with numbers, but SMT uses random words whereas NMT captures the similarity between words. SMT uses n-gram model and hence the context is always local. However, NMT uses RNNs and hence sentences which have long distance dependencies can be translated better using NMT. SMT consists of subcomponents that are separately optimized, whereas a neural network is optimized as a whole. NMT uses word embeddings for words and internal states, thus its model is simpler with just a single sequence model that predicts one word at a time. SMT, in contrast, has a separate language model, translation model and reordering model. The biggest benefit to NMT is its speed and quality. According to the analysis in [4], NMT outperforms SMT in BLEU and HTER scores. Also, NMT copes better with lexical diversity and makes less morphology and lexical errors than SMT. Because of these and other major advantages NMT has performed well compared to SMT.

	SMT	NMT
Core element	Words	Vectors
Knowledge	Phrase table	Learned weights
Training	Slow Complex pipeline	Slower More elegant pipeline
Model size	Large	Smaller
Interpretability	Medium	Very low Opaque translation process
Introducing ling. knowledge	Doable	Doable (yet to be done!)
Open source toolkit	Yes (Moses)	Yes (many!)
Industrial deployment	Yes	Yes (now at google, systran, wipo)

Fig. 1. SMT vs. NMT

On the other hand, NMT has a few problems as well. NMT can translate sentence by sentence, but when we consider the text as a whole, NMT fails to capture the document level context and only returns disjoint translated sentences. NMT also needs large amount of training data compared to SMT system. To attain better document level context attention mechanism was introduced in NMT and it achieved competitive results.

III. Data Collection and Preprocessing

The NMT model was trained on two separate datasets, one being the Open Subtitles dataset and the other being the more varied EU proceedings dataset.

- The Open Subtitles dataset has been obtained from public domain available at the following URL: http://opus.nlpl.eu/OpenSubtitles.php. The training set contains about 70,000 English sentences and corresponding 70,000 German sentences. The following operations were performed on text data:
  - Creating word vocabulary - Word vocabulary of most common 20000 English words and 30000 German words was generated using Counter and Map functions in Python.

  - Word tokenization - Further a word to index dictionary was generated by enumerating the most frequent words in descending order.
  - Creating fixed length sequence - RNNs need a fixed length sequence as input. Hence, sentences with length more than 15 were removed and data padding was performed on sentences with length less than 15 words.
- The Euro Parl dataset has been obtained from public domain available at the following URL: http://www.statmt.org/europarl/. The training set contains about 2 million English and corresponding 2 million German sentences. The following operations were performed on text data:
  - Creating word vocabulary - Word vocabulary of most common 100,000 English words and 100,000 German words was generated using Counter and Map functions in Python.
  - Word tokenization - Further a word to index dictionary was generated by enumerating the most frequent words in descending order.
  - Creating fixed length sequence - RNNs need a fixed length sequence as input. Hence, sentences with length more than 40 were removed and data padding was performed on sentences with length less than 40 words.

IV. NMT Model

- RNN/LSTM encoder and decoder: NMT model is based on encoder-decoder model for training. The input and output are both in the form of a sequence of words forming sentences. As such, it requires RNNs/Long Short Term Memory (LSTMs) on both sides to capture long term dependencies. Also, using LSTMs instead of vanilla RNNs is essential. While RNNs perform well in theory, they suffer from exploding and vanishing gradients. LSTMs overcome this weakness and capture long term dependencies better than vanilla RNNs. Hence, LSTM was used in the project as encoder and decoder.
- Embedding: Once fixed length sequences are generated, the data is fed into Tensorflow library that performs word embedding. This is the phase where words or phrases are mapped to vectors of real numbers and similarities between words is determined.
- Thought Vector and Projection Layer: While training the model, the intermediate outputs are vectors with a certain internal size. The final output of the encoder, however, is a single thought vector of the same internal size that encompasses all the information of a sequence required for translation. Thought vector and decoder inputs are both given as inputs to the decoder and the model is trained on large amount of data to generate another vector that contains the translated sentence. However, this vector needs to be converted from vector to word indexes or tokens. Projection layer performs this task when the translation is done.
- Attention Mechanism: Attention Mechanism allows the decoder to attend to different parts of the source sentence at each step of the output generation.

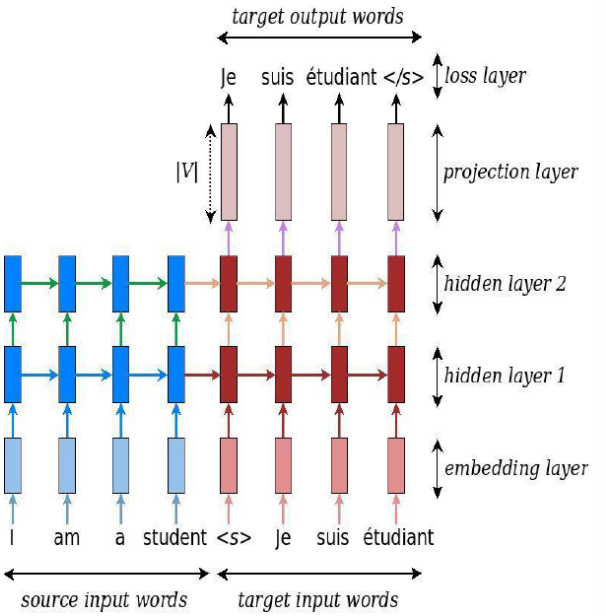

Fig. 2. NMT Model overview

V. Training

The following hyperparameters are used to train the model on Open Subtitles dataset:

- 1000 steps or epochs
- Batch size of 64
- Learning rate of 0.25

The following hyperparameters are used to train the model on Euro Parl dataset:

- 2500 steps or epochs
- Batch size of 64
- Learning rate of 0.25

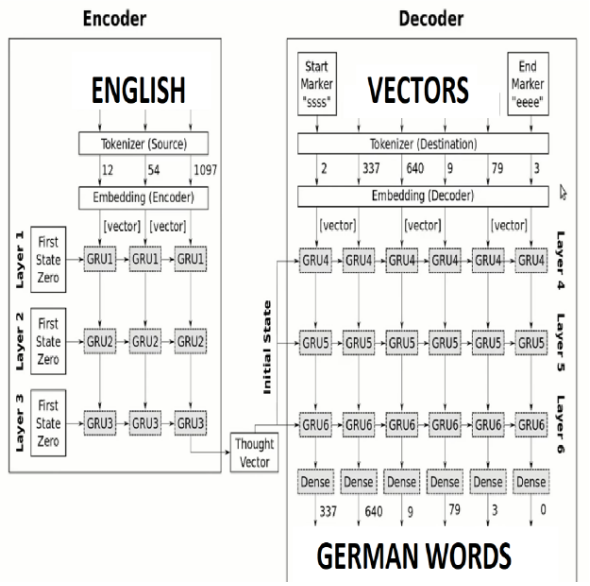

Fig. 3. NMT Model detailed

These hyperparameters were chosen after going through several trials. Softmax function is used as the loss function here whereas optimization is performed using RMSPropOptimizer. Tensorflow library provides a convenient way to define loss and optimizer functions, and hence it's easier to focus on solving the problem at hand rather than delving into building loss and optimization functions. Figure 3 shows the training loss plotted every 5 training steps. It is clear that the training loss reduced exponentially over time and was down to 1.26 units after 1000 training steps. Here loss is calculated from the optimizer function.

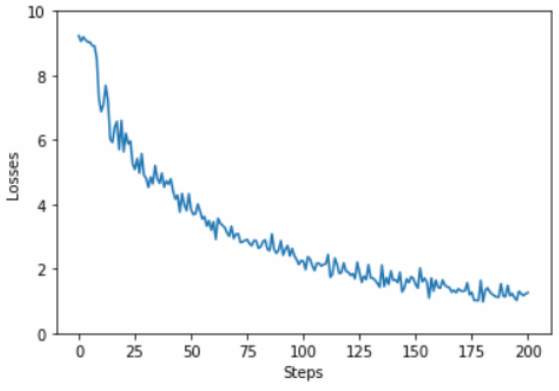

Fig. 4. Training loss after each step

VI. Results

Following results were obtained when the model was tested on conversational English sentences. These results are shown in Fig. 4. To test the validity of results obtained from the trained model, they are compared with Google translate for reference as shown in Fig. 5 and Fig. 6.

If we compare the results obtained from trained model and the results obtained from Google Translate, it is evident that except for sentences 2 and 7 almost all other sentence translations are comparable to translations obtained from Google Translate.

This model can be improved further by training it for a longer period of time, having a larger dataset, and/or by better selection of hyperparameters (i.e. learning rate, batch size).

The model trained on Euro Parl dataset was further evaluated formally using the BLEU algorithm. It was tested on 15 sentences with BLEU score calculated for each sentence. Average BLEU score of all the sentences was found out to be 0.71.

VII. Conclusion / Future Scope

The results obtained in this project signify a few things: 1) With only 30 minutes of training on a personal computer, a model can learn to translate short sentences from English to German with considerable accuracy. If such a naive model can get such good results, focused research can certainly make significant breakthroughs. This suggests that NMT has great untapped potential and that we have yet to realize the full potential of NMTs. 2) As discussed earlier, longer sentences can generate incorrect translations and suffer from lack of context. This is visible in results obtained where sentences 2 and 7 weren't translated correctly. Larger context NMTs for extremely long sequences like paragraphs, articles, books can be achieved by having effective attention mechanisms. 3) By combining the results from a machine translator with another system like summarizer, dialog system, or speech recognition system, the potential applications in real life are extremely versatile. We can have many systems working in tandem as we only change the mode of input from text to audio or audio to text but the basic storage mechanism stays the same. Thus, these architectures are generic and have the same basic backbone of implementation. 4) All of this leads to the conclusion that these ideas and approaches are

```
1.
--------------------------------
What' s your name
Wie heiÃŸt deine Name
--------------------------------
2.
--------------------------------
My name is
Meine ich bin
--------------------------------
3.
--------------------------------
What are you doing
Was machst du da
--------------------------------
4.
--------------------------------
I am reading a book
Ich bin ein Buch
--------------------------------
5.
--------------------------------
How are you
Wie geht' s
--------------------------------
6.
--------------------------------
I am good
Ich bin gute
--------------------------------
7.
--------------------------------
Do you speak English
Haben du das reden
--------------------------------
8.
--------------------------------
What time is it
Was ist es ist
--------------------------------
9.
--------------------------------
Hi
Hallo
--------------------------------
10.
--------------------------------
Goodbye
Wiedersehen
--------------------------------
11.
--------------------------------
Yes
Ja
--------------------------------
12.
--------------------------------
No
Nein
```

Fig. 5. Test data results

going to revolutionize human computer interaction in unexpected and astonishing ways in the near future.

VIII. Acknowledgements / Tools

I would like to thank Prof. Matthew Stone for his constant support and guidance in this project.

Here is the list of libraries used in this project:

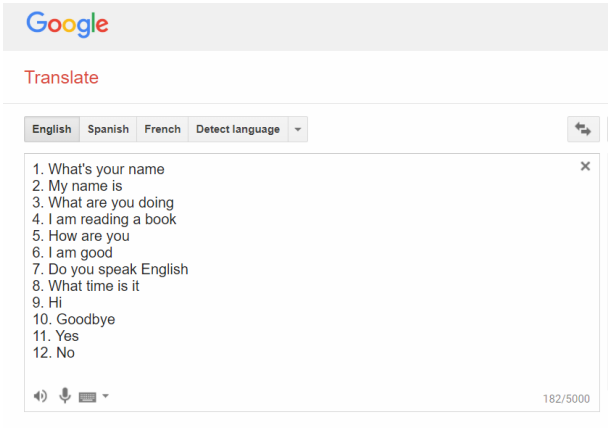

Fig. 6. English input to Google Translate

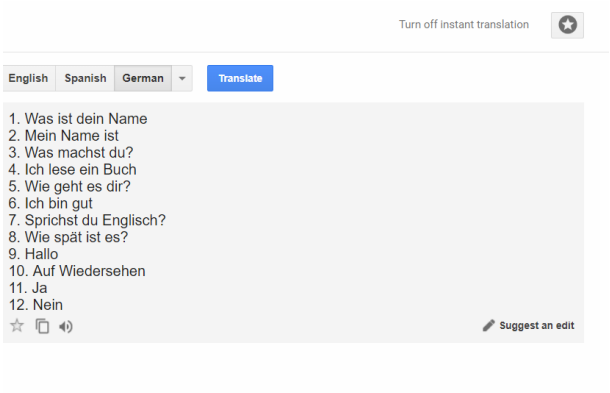

Fig. 7. German output from Google Translate

- TensorFlow to build machine learning model and test the model
- Numpy to write helper functions for building machine learning model
- Matplotlib for data visualization
- Scikit-learn to split training and testing data

IX. References

[1] Denny Britz, Anna Goldie, Minh-Thang Luong, Quoc L (2017). *Massive Exploration of Neural Machine Translation Architectures*. Google Brain.

[2] Minh-Thang Luong, Hieu Pham, Christopher D. Manning (2015). *Effective Approaches to Attention-based Neural Machine Translation*. Computer Science Department, Stanford University.

[3] Philipp Koehn (2005). *Europarl: A Parallel Corpus for Statistical Machine Translation*. MT Summit 2005. Retrieved from http://www.statmt.org/europarl/

[4] Herv Blanchon, Laurent Besacier. *Comparing Statistical Machine Translation (SMT) and Neural Machine Translation (NMT) Performances*. Laboratoire LIG quipe GETALP. Retrieved from http://lig-membres.imag.fr/blanchon/SitesEns/NLSP/resources/SMT-vs-NMT.pdf

[5] Martin Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng (2016). *TensorFlow: A system for large-scale machine learning*. Google Brain.

[6] Jrg Tiedemann (2009). *News from OPUS - A Collection of Multilingual Parallel Corpora with Tools and Interfaces*. In N. Nicolov and K. Bontcheva and G. Angelova and R. Mitkov (eds.) Recent Advances in Natural Language Processing (vol V), pages 237-248, John Benjamins, Amsterdam/Philadelphia.

[7] Bojar, Ondrej; Chatterjee, Rajen; Federmann, Christian; Graham, Yvette; Haddow, Barry; Huck, Matthias; Yepes, Antonio Jimeno; Koehn, Philipp; Logacheva, Varvara; Monz, Christof; Negri, Matteo; Nvol, Aurlie; Neves, Mariana; Popel, Martin; Post, Matt; Rubino, Raphael; Scarton, Carolina; Specia, Lucia; Turchi, Marco; Verspoor, Karin; Zampieri, Marcos (2016). *Findings of the 2016 Conference on Machine Translation*. ACL 2016 First Conference on Machine Translation (WMT16). The Association for Computational Linguistics: 131198.

[8] Dzmitry Bahdanau; Cho Kyunghyun; Yoshua Bengio (2014). *Neural Machine Translation by Jointly Learning to Align and Translate*. ICLR Conference Paper

[9] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu (2002). *BLEU: a Method for Automatic Evaluation of Machine Translation*. Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), Philadelphia, July 2002, pp. 311-318.

[10] Philipp Koehn (2017). *Neural Machine Translation*. Retrieved from http://mt-class.org/jhu/assets/nmt-book.pdf

[11] Siraj Raval (2017). *How To Make a Language Translator*. Retrieved from https://github.com/llSourcell/How_to_make_a_language_translator

[12] Wok, Krzysztof; Marasek, Krzysztof (2015). *Neural-based Machine Translation for Medical Text Domain. Based on European Medicines Agency Leaflet Texts*. Procedia Computer Science. 64 (64): 29. Retrieved from https://ac.els-cdn.com/S1877050915025910/1-s2.0-S1877050915025910-main.pdf?_tid=1ab9e183-3c35-426b-bcee-0226db691b64&acdnat=1537230761_b54616ae04f6c548f691a6bc54d63ad0