Machine Learning Advancements

Majority of the today’s challenges with machine learning revolve around the insurmountable about of data that needs processing. However, language data has a different issue, its main issue is the lack of data. Language data is scarce and sometimes nonexistent. Google has been trying to tackle this issue because majority of their products help us communicate with each other through apps on our phones, smart home devices, and other products. The challenge has been to make all their products available throughout the world on many different platforms with support for different languages.

There are generally two traditional approaches to language translation with the first being machine translation. The challenges of machine translation are that there are so many rules for different languages. Because of the natural ambiguity and flexibility of human language, each language may operate at different lexical, syntactic, or semantic levels that makes it very difficult for one machine model to be truly universal. The ability to transfer results to another language without faults will be a difficult feat to accomplish. Not only does this require specialists to aggregate all the rules for a language, but engineers need to become familiar with them as well.

The second approach is called statistical neural translation. Statistical neural translation approach tries to determine the highest probability of generating a good translation based off a statistical model. This model quickly outperforms a machine translation model but comes nowhere near neural machine translations models in terms of performance. This model works particularly well with short phrases but fails on larger texts which cannot be discounted in the translation possibilities.[5]

The latest and state of the art in language processing utilizes neural machine translation. Google has named their version Google NMT. Over the years, Google has done enormous strides with their neural machine translations. They have been striving to adopt a single translation model based on the notion that “the learning signal from one language should benefit the quality of translation to other languages.” [1] Neural machine translation uses neural networks to learn a statistical model for machine translation. The model can learn on input and output text directly, thus removing the requirements for rules of each of the languages. The primary advantage of a neural machine translation is that it is truly end to end. Cons of neural machine translation systems however include slow speed of training the models, ineffectiveness in dealing with rare words, and sometimes failure to translate all words in a sentence. The following figure helps one visualize the input and output capability of NMT which is commonly known as an encoder decoder process:

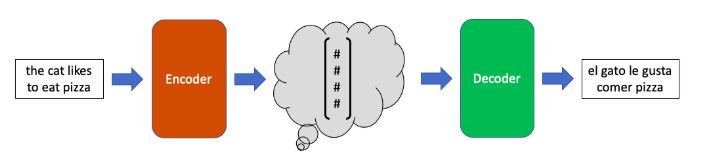


Figure 1 Encoder/Decoder [2]

The goal of this universal neural machine model is to be capable of translating between any language pair. On languages with very little training data, performance is improved on the languages with very little training data due to positive transfer. But what is positive transfer? Positive transfer learning is a machine learning technique where a model trained on one task is re-purposed on a second related task. However, the cons of this are that languages with extremely high amount of data tend to get worse. Google has found that the more languages added to this universal model, the quicker the quality drops.

As a result, while a universal NMT may be good, it is still in its infant stages. Even though Google’s NMT performs 60% in the best-case scenario than its phrase-based implementation used in its current applications, it is still very computationally expensive. Thus, this suggests that developers might want to keep their strong baseline languages and use the universal NMT to be a catch all solution for the other languages which lack data. The amount of the data used by Google in this research endeavor is on the order of 25 billion sentences. [4] This was achieved by crawling and extracting parallel sentences from the web. Because all the sentences were extracted from the web, any lack of data in training set would require serious research work to resolve because this data obtained was actual data by people and not generated. The follow two figures show the performance of GNMT in action:

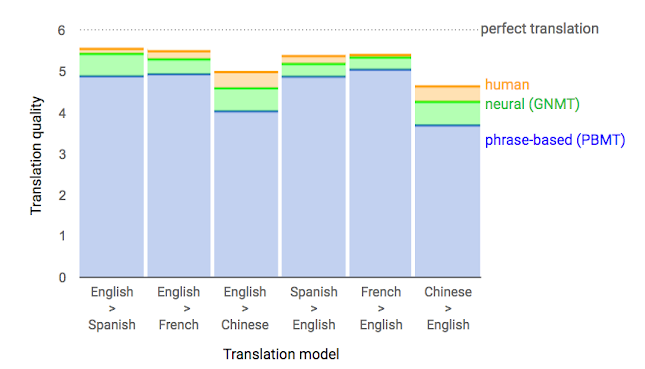


Figure 2 Google NMT comparison [6]

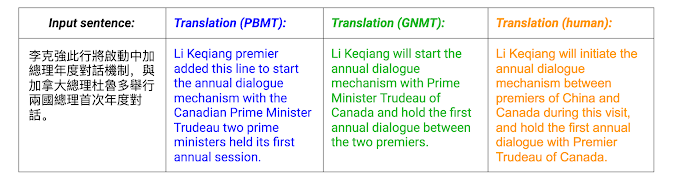


Figure 3 Translation Snippet [6]

One can clearly see that the phrase-based translation is not as fluid as the GNMT implementation. Also, the translation quality of the GNMT is about a point higher in translation quality with translations also of that of a human translation.

The following figure is an open source project to demonstrating how an NMT implementation works:

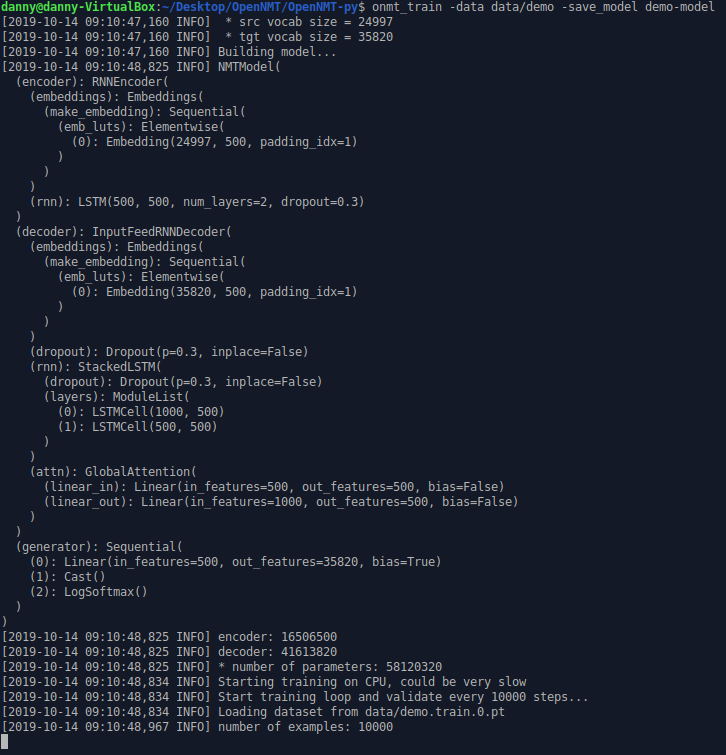


Figure 4 OpenNMT Example [3]

In this opensource version of NMT, training can be run on the GPU or CPU. The example above is done on a virtual machine via CPU. This opensource project relies on Pytorch which in tun relies on TensorFlow. If a user wants to evaluate OpenNMT, they should preferably have a dedicated Linux machine with a GPU having CUDA cores. Windows machines will not have the necessary required packages to run this project. To train the example training set and run through the example classification would take several days. This shows how resource intensive NMT currently is and how it is not ready for portable devices.

Ultimately, while the intention that Google has is good, there is still a long way to go. This is a project research over 5 years in the making and still will not be reaching maturity for several years. Over half of 7,000 languages will no longer be spoken by the end of the century. If Google succeeds, they hope to preserve these languages with this universal neural machine translation model.

References

[1] Exploring Massively Multilingual, Massive Neural Machine Translation. (2019, October 11). Retrieved from <https://ai.googleblog.com/2019/>

[2] Neural Machine Translation. (2019, June 3). Retrieved from <https://towardsdatascience.com/neural-machine-translation-15ecf6b0b>

[3] OpenNMT. (n.d.) Retrieved from <https://github.com/OpenNMT/OpenNMT-py>

[4] What’s so Massive About Google’s Massively Multilingual Neural Machine Translation? (2019, July 18). Retrieved from <https://slator.com/technology/whats-so-massive-about-googles-massively-multilingual-neural-machine-translation/>

[5] Introduction to Neural Machine Translation. (2019, August 7). Retrieved from <https://machinelearningmastery.com/introduction-neural-machine-translation/>

[6] A Neural Network for Machine Translation, at Production Scale. (2016, September 27). Retrieved from <https://ai.googleblog.com/2016/09/a-neural-network-for-machine.html>

Group Summaries

Lian, Yunze

In Yunze’s report, he/she reported on Fast R-CNN, R-CNN and CNN. I learned that R-CNN in comparison to CNN performs better and Fast R-CNN performs quicker than traditional R-CNN at the cost of accuracy. The way R-CNN works is that there are generally around 2000 region proposals of interest. These regions are then fed to the classification and localization network followed by an SVM machine for the final classification. CNN is used regularly in computer vision to detect objects in images. I have learned from Yunze that detecting multiple objects is computationally more expensive than detecting a single object through CNN. Ultimately, they all suffer from the same issue in that they are very computationally expensive.

Hu, Xushan

In Xushan’s report she talked about the various ways to detect and segment objects. The following are the types detection and segmentation methods used commonly used: Support Vector Machine (SVM), Mask-RCNN, Region-based Fully Convolutional Networks (R-FCN), and Single Shot Multibox Detector (SSD). SVM maps data to a high dimensional space and finds the hyperplane to maximize margins between classes. I learned that it is extremely efficient and robust in content-based image classification. Mask R-CNN is based off fast R-CNN and adds a little bit of overhead to an otherwise fast but less accurate segmentation method.

I also learned about R-FCN. From the report it states that it has higher accuracy and faster operation speeds compared to other neural networks. This framework can be divided into two sublayers: a shared subnetwork and a subnetwork without sharing computation. I learned that these layers are convolutions and shareable in the whole picture to help with object detection.

Lastly, Xushan mentioned an opensource implementation called TensorFlow Zoo. It has a lot of capabilities for visualization but is apparently slow. Recommendations on using TensorFlow Zoo requires a high-performance GPU to do the training. A multi-threaded CPU can be used to queue data for processing.

Zhao, Peixi

In Peixi’s report, he gave a summary of the various types of unsupervised learning methods used today. The report listed the following commonly used algorithms: hierarchical clustering, k-means clustering, density-based spatial clustering of applications with noise DBSCAN, mean shift and more. I learned from the report that the challenges in unsupervised learning is to find some sort of structure in the input because the data itself is no labeled.

There were also a couple of examples of unsupervised learning with k-means and CNN. K-means is simple and easy to implement with the ability to detect what the human eyes cannot see. One of the examples was an experiment to determine if there was a relationship between working behaviors and the stress of people. Unsupervised learning on the data set found the relationship of each cluster to a stress level.

In conclusion, unsupervised learning’s success is driven by context of the data. Knowing what to expect will aid in the classification of unsupervised data.

Zhu, Chenhui

In Chenhui’s report, he/she gave an evaluation of the overall success of machine learning over the past few years. Deep Convolution networks became more signification for recognition with “AlexNet” being the first notable architecture. There were other structures mentioned in the report such as the following: VGGNet, ResNet, and Inception-Net. There was also mentions of the applications of image segmentation. They are primarily used in robot vision, autonomous driving, indoor navigation, and medical applications in today’s world.

Chenhui focused on the Deeplab architecture in detail. He/she mentioned that pooling and downsampling in traditional CNN classification leads to a decrease in spatial resolution. DeeplabV3 uses an encoders and decoders which apparently tries to ameliorate this issue. The report concluded that DeeplabV3 is the best architecture in terms of segment detection compared to other models on the PASCALVOC 2012 semantic image segmentation benchmark.