* Problem Statement:

**Transformer model with Self-Attention Mechanism - Bidirectional Translation**

While attention mechanism is an interesting discovery to assist decoders in neural machine translation model, self-attention mechanism further develops the idea and get rid of the limit of sequential model to allow us to make the best use of parallel processing computing power and gives us decent results.

Through this project, the author aims to explore how well a machine learns by comparing its translation back and forth. Specifically, I trained two transformer models, one to translate English to French, the other to translate French to English with the same data.

* Configurations of Hardware:

The training is through Google Colab with Python 3 Google Compute Engine Backend GPU with 25GB RAM and 360 GB disk (note that the disk size could be smaller, but the RAM must be at least 25 GB).

It took around 2 minutes for 100 steps, each epoch includes 1300 steps and only 3 epochs are trained due to such limited computational resource. The total training time for the two models is about 2 hours.

* Configurations of Software:
  + Pytorch 0.3.0 for Linux 64 Bit machine
  + Torchtext 0.2.3
  + Fastai 0.7.0
  + Torchvision 0.1.9

Note that the versions of all the listed packages much strictly match for compatibility.

* Dataset:

IWSLT French to English Data; SpaCy English and French Language Resource[[1]](#footnote-1)

* Results:

A good model for French to English but a bad one for English to French.

* What Worked, What not Worked and Lessons Learnt

The project is based on the great post at <http://nlp.seas.harvard.edu/2018/04/03/attention.html>. While it’s a breakdown for the paper *Attention is All You Need*, it fails to keep updated with the recent Pytorch version and actually doesn’t work. The actual post time is in April, 2018, more than 1 year ago. This led to a great effort for testing and debugging the code or finding the right version of all used packages in order for the script to run smoothly. Some lines are missing and some posted blocks couldn’t run.

For example, there aren’t an attribute of *encode* or *decode* for class *EncoderDecoder*, which lead to runtime error during training; in the later section when giving an translation example, the original posted code was *start\_symbol=TGT. stoi["<s>"])* , which gave the error saying *TGT* doesn’t have a method *stoi*, the author has to add *.vocab* to make it work.

Since the paper and the model is so important, a lot of developers have complained on similar problems and I was able to follow their solutions and worked it out. A valuable lesson learnt through this process is to document whatever code well, especially the versions of the packaged used and the hardware configurations to make it reproducible.

* Youtube URL: <https://youtu.be/Lp79IAF1XFQ>

1. <https://wit3.fbk.eu/download.php?release=2017-01-trnted&type=texts&slang=fr&tlang=en>  
    <https://github.com/explosion/spacy-models/releases/download/de_core_news_sm-2.1.0/de_core_news_sm-2.1.0.tar.gz#egg=de_core_news_sm==2.1.0>

   <https://github.com/explosion/spacy-models/releases/download/fr_core_news_sm-2.1.0/fr_core_news_sm-2.1.0.tar.gz#egg=fr_core_news_sm==2.1.0> [↑](#footnote-ref-1)