MÜ – Übung 5 **Tannon Kew**
Encoder-Decoder Model

For this exercise, I used the supplied Europarl corpora files to train an encoder-decoder translation model from German to English.

Preprocessing

In a first step, the data was preprocessed according to the recommended steps for preparing data for a machine translation system. This included first normalizing and tokenizing the corpora data with the help of the scripts supplied by the Moses Decoder Machine Translation System. Following this, true-casing was trained on the two training sets and then applied, also using Moses. In a final preprocessing step, a BPE model was trained on the training sets for each language, both using 80,000 symbols each. The trained BPE models were then applied to all corpora files for the relevant language.

Changes to Daikon

In order to improve the performance of the translation system, I then implemented a couple of small changes to the Daikon code. An early stopping function was introduced, designed to automatically quit the training process, once perplexity on the dev set showed a consistent increase after 5 iterations. Additionally, the source language input sequence was reversed, as suggested by Sutskever (2016). The motivation behind this change was to reduce the long-term dependencies between first few words in the input sequence and the first few words produced by the decoder. After counting the types appearing in the source and target language training sets, I set the both vocab size variables to 70,000 in order to cover as many of the most common words in each language as possible. As a last alteration, I aimed to supress the model's production of unknown words in the translation module. With this change, if an unknown word is the most probable word in the sequence, the model will produce the second most probable word.

Training Parameters

Finally, training was initialised, specifying the maximum number of epochs as 10 and enabling the option to produce a translation after each training epoch.

Evaluation and Postprocessing

Training was finished manually after 5 epochs. The early stopping function did not exit the program within the training time as the dev set perplexity never actually increased from the previous perplexity score 5 consecutive times. Rather, at the time of stopping the training, it hovered around 3.15.

After using the trained model to translate the source language dev and test sets, these translations were scored with daikon's scoring function. The average corpus perplexity on these files was 1.11. Using the BLEU script from exercise 1, translated, postprcoessed dev set received a BLEU score of 22.39 using the supplied English dev set as the reference.