MT Exercise 2

Authors: Zainab Aftab & Kristina Horn

1. Cleaning Parallel Training Data

1. *Eyeballing the cleaned parallel data, what is your impression? Does the filtering work well?*

   Looking at the output files, it seems that the filtering does indeed work. We were surprised to see that when the same line in the input source file as well as the input target file doesn't have a sentence in the same language, the sentence is then completely filtered out. That means that the output files don't contain the sentence anywhere. To get a cleaned or filtered file this is a beneficial method but it is important to note that the removal of sentences leads to content being lost.

2. *How do different character ngram orders influence language identification accuracy, and by extension filtering results?*

   How effective different ngram orders are for different languages, depends on which language we use. Lower n-gram orders tend to be better at identifying languages with distinctive character distributions, like Chinese or Korean. Those languages have a rather small number of distinct characters and the distribution frequency is quite distorted, because only a few characters account for a large proportion of the text. That's the reason why lower n-gram orders are more effective.
   If we look at trigrams or 4-grams, these n-gram orders are better at detecting European languages, which have less distinctive distributions.
   Higher n-gram orders however lead to high risk of overfitting even more if the training data is limited.
   Overall the choice of n-gram order depends on the characteristics of the language that is being analysed and the goal of the analysis.

   If we look a bit further and have a look at how the character n-gram order influences filtering results, we will get some similar results. The choice of n-gram order can have a significant impact on the accuracy and efficiency of filtering systems. Higher n-gram orders capture more detailed and subtle patterns but they also increase the dimensionality of the feature space and can lead to overfitting. Lower n-gram orders may miss some of the finer-grained patterns in the text, but they also lead to a m ore compact and robust feature representation. So it is advisable to use a combination of different n-gram orders to improve the overall performance of filtering systems.

3. *Can you think of a way to also filter out the surprise language, without having a trained language model in this language?*

   One option could be to download a package that is able to identify the

language like *langid*. Libraries like *langid* identify the language of a given string and compare it with a list that contains the language abbreviations (e.g. ‚it', ‚en' or ‚de') of expected languages.
A second way to filter out the surprise language is to create a list of common characters that appear in a language. If the input text contains characters that can be found in the list, then you have the surprise language.
Lastly, the nltk libary lets you create a list of common stop words of a specified language. You can check if the input text contains any of the stop words and if that is not the case, you can assume that you have encountered a surprise language.

2. Impact of Postprocessing on Translation Quality

Postprocessing steps	sacreBLEU settings	BLEU
Full		15.0
Full but lowercased	-lc	20.3
Tokenise with Moses tokenizer instead	--tokenize none	9.6
Tokenise *hyp* and *ref* differently	--tokenize none	98.8

1. *Which method achieves the highest BLEU scores? Why?*

   The method with two different Tokenizers for the hyp and the ref produces the highest BLEUScore. It's a bit difficult to justify why this happens, because we don't really know how the two tokenizers work in depth. So we can only assume… We think it could be caused by smaller generated tokens from one tokenizer. If we have smaller units there can be more matches between the hypothesis and reference text. Because of this the BLEU score can be artificially inflated.

2. *Following up on the last question, what are some flaws in the translation setup we used here? How could it be improved*

   Because we trained on a smaller data set, the translation may be not as accurate as expected. Additionally the JoeyMT model may be sensitive to the quality of the training data. If the training data is not representative or of poor quality, the performance of the model may be suboptimal, which could also affect the BLEU score.
   To improve those flaws you cloud use multiple translation models and compare

their results to get an even deeper insight and reduce the influence of the model on our hypothesis about the BLEU score. You also probably should use a larger and more diverse training dataset to really receive meaningful numbers and hypothesis.

3. *Describe the effect of using two different tokenizers and how this relates to your observations in the BLEU score.*

   Using different tokenization methods can affect the calculation of the BLEU score in many different ways. Different tokenizers can for example result in different number of tokens in the hypothesis and reference text. This can effect the n-gram matches and the brevity penalty calculation, which will then affect the overall BLEU score. If one tokenization method produces more or fewer tokens than the other, the comparison between the hypothesis and reference texts becomes less meaningful. For example, if the hypothesis text is tokenized using a method that produces more tokens than the reference text, the BLEU score might be artificially inflated.

4. *Describe what an ideal standardized way to report BLEU scores in a research paper should look like in your opinion.*

   Ideally, postprocessing steps should be made transparent in research papers as the BLEU scores can vary depending on the steps taken. First, the tokenizer used and the number of tokens made should be specified. Different translations are produced if a different tokenizer was implemented and therefore potentially changed the number of tokens generated. Furthermore, lowercasing the tokens is important to know as lowercasing can reduce the vocabulary size. In this case ‚car' and ‚Car' are treated as the same token whereas not lowercasing will treat it as different tokens. Lastly, research papers should indicate if different tokenizers were used for hyp and ref as it affects the results.

5. *How might the results have looked like if we used the opposite translation direction? Try to support your thoughts with good reasons.*

   Because German and English have dissimilar grammatical structures and different word ordering, the BLEU score may vary if we had used the opposite translation direction. Furthermore, the same tokenizer may be suited for a particular language and less suitable for others. As a result, a better BLUE score will be calculated. Next, the quality of the available translation may influence the evaluation. For example, if the quality of the German to English translation is better than the translation from English to German, then even using the same tokenizer could result in a different BLUE score.