German to English Machine Translation

Brandon Pokorny
Linguistics
University of Illinois
pokorny3@illinois.edu

Abstract

I describe the technniques used to create a machine translation system for translating from German to English. The system is focused primarily on news text with measures to handle more robust text that may be encountered. The system uses data cleaning through language recognition, subword tokenizers, sentence pair length measurements for data filtering, back translation, adaptive fine-tuning, ensembling, and light post-processing to accomplish its goals. It also aims to use knowledge distillation, reranking, additional placeholders to handle the more robust text, and create and use a synthetic parallel corpus. The model will be Transformer Big neural nets built with OpenNMT-py, which will also help with a few of the techniques above. This paper will detail how and why those techniques are used, how the performance is tested, and how they perform, which will be measured via a BLEU score.

1 Introduction

This paper describes the techniques I used to build a machine translation system for translating German into English, as well as the results of that system. These two languages are not too different from each other when it comes to spelling and some words that are definitely borrowed, but ultimately pose a challenge especially when the sentences need to be reordered.

According to Naver Labs Europe's submission to the WMT 2019 Shared Task (Bérard et al., 2019), data cleaning earned a vast majority of their points so I attempt to tackle this area thoroughly. As one common issue encountered is miscellaneous languages showing up in the samples, I first ensured that the data is clean using the python langdetect module; any sentence pair that does not have the correct languages on the correct sides will be removed. I also ensured that data with words too large or too uneven character ratios between the sentence pair, as in (Li et al., 2019), are removed from the training data. Additionally, I replaced emojis, numbers, and emails with placeholders as done successfully by Naver Labs Europe (Bérard et al., 2019). Since these sequences have an unchanged meaning in both languages, they don't necessarily need to be translated so much as just persist through both sides of the translation.

In order to augment the size of my data, I also back-translated monolingual text to create additional data to train with using a pre-trained English to German model offered by OpenNMT-py (Klein et al., 2017). Since tokenizing and BPE are absolutely required for competitive performance in recent systems (Bojar et al., 2019), I used SentencePiece's ULM (Kudo, 2018) to preprocess and standardize my data as well before feeding it into my model.

The final models are 3 Transformer Big models made with OpenNMT-py and ensembled as very few systems did not take advantage of ensembling. In training each model in the ensemble, I will also have a separate dataset for adaptive fine-tuning to ideal news sentence pairs. Finally, there was some trivial post-processing performed in the event that German punctuation persists in the English ouput; while English and German share most of their punctuation, German does have a second set of quotation marks absent in English and it is a simple fix to swap any German quotation marks with English quotation marks should they appear in the output.

If time allows it, I would like to replace any anomalous language instances in the source side of sentence pairs with placeholders instead of removing the entire sample since sometimes oter languages would show up in text to be translated. I also would want to expand emoji placeholder replacement to emoticons with japanese characters, which

can be recognized with the nagisa python module as used by NTT (Murakami et al., 2019). In order to further augment my data, I would like to create additional synthetic data with unknown words as demonstrated by PROMT's submission to WMT19 (Molchanov, 2019). Finally, since n-best reranking and knowledge distillation were very popular as well, I would like to implement these in my final system as they both had positive effects on the BLEU scores of various systems. NiuTrans offers a detailed description of their n-best feature reranking method as well as their knowledge distillation in their system (Li et al., 2019) that I think would improve my system too.

2 Prior Work

The following outlines the current state-of-the-art techniques recently used for the task of translating news from German to English.

2.1 Preprocessing Text

Preparing the data is a huge part of the task that is often considered crucial to the success of the system. A significant portion of the submissions to the WMT19 used MosesTokenizer to normalize the punctuation (Iranzo-Sánchez et al., 2019), add truecasing (Iranzo-Sánchez et al., 2019) as well as tokenize the text (Ng et al., 2019). Difficult characters would also be removed like "some special UTF-8 characters" (Rosendahl et al., 2019). Afterwards, BPE is very frequently performed for subword segmentation and is usually joint BPE with merge operations ranging from roughly 35k-50k, often done so with OpenNMT (Molchanov, 2019) or SentencePiece as in (Bawden et al., 2019). MLLP also further prepares their BPE by removing from their vocabulary the subwords that appeared less than 10 times (Iranzo-Sánchez et al., 2019). However, there were a few examples that deviated in their BPE strategies: RWTH decided to experiment between the more popular joint BPE approach and the unigram language model approach also offered by SentencePiece (Rosendahl et al., 2019), which did achieve a higher BLEU score using a 30-best list and 50k vocabulary. Another important finding from PROMT is that case insensitive mode for BPE is the cause for a noticeable increase in performance, which motivated their preference for OpenNMT's BPE over Marian's BPE (Molchanov, 2019). Placeholders and tokens also saw use. PROMT replaced numbers or alphanumeric sequences with placeholders as these do not need to be specifically translated (Molchanov, 2019).

2.2 Data Cleaning

Cleaning and filtering the training data also had a great impact on the performance of the systems. Many teams in the WMT19 found the data from ParaCrawl to be very noisy and overall hurt performance without careful filtering. There were a few different ways to do this and while MLLP (Iranzo-Sánchez et al., 2019) tested the use of language models to filter out harmful sentence pairs, some teams such as CUED also adopted popular heuristics to successfully filter the training data (Stahlberg et al., 2019). These heuristics include language detection, where they removed sentences from both sides if it did not match the intended language, removing sentences with words that had above 40 characters, removing sentences with HTML tags, removing sentences with less than 4 words, requiring sentences to be equal after removing non-numerical characters, removing sentences with a character ratio between the pair larger than 1:3 or 3:1, and removing sentences that didn't end in punctuation.

The idea of filtering based on sentence lengths or character ratios also showed up in other systems as well. NiuTrans had similar heuristics in that they filtered sentences that had a word with over 40 characters or the sentence itself contained over 100 words (Li et al., 2019). They also had an identical character ratio rule for filtering, removed sentences with HTML tags, and used language detection to filter sentence pairs in which both sides were the same language. As for acertaining the quality of the pair, they filtered out sentence pairs whose alignment scores were below 6 with a fast-align toolkit.

PROMT also filtered out anomalous languages by using an ensemble of langdetect, pycld2, and langid, took steps to remove duplicated translations, and employed length and ratio heuristics (Molchanov, 2019). For removing the duplicated sentences, they looked primarily at the source side for duplications and kept only the target side translation that had the highest frequency. On top of that, lines with rare words were filtered out. Rare words were determined by "frequency lists built on large monolingual corpora" (Molchanov, 2019).

2.3 Data Augmentation

Backtranslation from monolingual data was extremely prevalent so as to increase the training data. Usually this involves training a transformer model to backtranslate monolingual data as PROMT did (Molchanov, 2019). The monolongual data would be in the target language with backtranslations provided for their source language counterparts, thus creating parallel data for usage in training (Albeit not as perfect as most provided parallel data). However, PROMT also managed to replicate their data by subsituting unknown words. They'd use the fast-align tool to identify aligned words in sentence pairs and randomly subsitute 1-3 certainly aligned words with a <UNK> placeholder. The goal of this was to train the model to preserve this placeholder in the translation, which was useful for handling named entities recognized in their system.

2.4 Knowledge Distillation

Knowledge distillation is another technique that showed up in a few systems, such as NiuTrans' system (Li et al., 2019). Multiple student models are ensembled together to make a teacher model, which in turn imrpoves the student models using source data. Multiple iterations of this are performed until the increase in BLEU score is negligible, but unfortunately the dificiency of diversity held this method back in attaining good results. To counter this, different teachers were assigned different students in each iteration, creating more diversity and therefore improving the BLEU score more.

2.5 Reranking

Many systems additionally employed n-best reranking including NiuTrans. They used a variety of different models to rescore 96-best translations from several ensembles (Li et al., 2019).

2.6 Model Design

Most models use Transformer Big models, like Facebook Fair (Ng et al., 2019). Transformer Base models trained for German to Englishwere very rare, but ensembles of Transformer Big models were much more prevalent. Popular toolkits for building the models were Sockeye (Rosendahl et al., 2019), MXNet (Rosendahl et al., 2019), fairseq (Ng et al., 2019), MarianNMT (Molchanov, 2019), and PyTorch (Xia et al., 2019). Regarding those models, the popular optimizers were Adam optimizer (Ng et al., 2019) and the MultistepAdam optimizer from Tensor2Tensor (Stahlberg et al., 2019).

2.7 Post Processing

Not a lot of post-processing is done, but Facebook Fair did replace / correct punctuation marks in the post-processing (Ng et al., 2019). This served as an easy way to see a small increase in BLEU score because German input sometimes uses different punctuation than English text does and correctly fixing any German punctuation found in the English output is a trivial task.

3 Approach

The following is the data that was used, how it was processed, and the method in which the translation system was built. All scripts can be found at https://github.com/Clickedbigfoot/ling506-project-scripts. All the data that was used was provided by the WMT organizers, ranging from parallel data in the source and target language to monolingual news data in the target language from which additional parallel data was created through back translation. After preprocessing the data and, in some cases, careful data filtering, models were trained using OpenNMT-py to fulfill the end goal of German-English translation.

3.1 Data

From the provided datasets, parallel data was drawn from Europarl, ParaCrawl, and CommonCrawl. Monlingual news data for German and English was drawn from NewsCrawl. As a first step, all unprintable characters were removed while certain sequences like emails or usernames were replaced with special tokens in both sides of the parallel data to ensure that they persist unadultered through the translation pipeline. Since ParaCrawl and CommonCrawl were often regarded as having low quality data by other teams in the WMT to the point that it is actually harmful to systems' performance, they also underwent special data filtering: the python langdetect module was used on the parallel data from those datasets to ensure that the source side was in German while the target side was in English. The sizes of translation samples were also closely examined; if a sentence contained less than four words, more than one hundred

words, or had a word that contained over forty characters, then that sample would be left out of the final data. Since SentencePiece tokenized words into subword segments, MosesTokenier wasn't a part of the tokenizing process, but it was still useful for counting the words in attempts to filter out data by length and length ratios. I also made sure to remove samples where the number of characters in the two sides of the translation exceeded a 1:3 or 3:1 ratio.

The next step for all data after tokenizing and any data filtering was to conduct subword segmentation with SentencePiece's unigram model. To do this, I used two separate SentencePiece models: one trained on the monolingual English data from NewsCrawl and the other trained on the monolingual German data from NewsCrawl. Both were trained with a vocab size of 23,000.

3.2 Method

The translation models used in this translation system are built using the OpenNMT-py toolkit. In order to augment the training data with more parallel data, I used a pretrained Transformer Base model offered by OpenNMT-py to backtranslate the German monolingual data from NewsCrawl and ran it through the same processing and filtering as the original three datasets. I then split the all the data into training, validation, and testing sets. The training sets consisted of 10,000 sentence pairs from Europarl, ParaCrawl, CommonCrawl, and the backtranslations from NewsCrawl each, whereas the validation and testing sets each consisted of 1000 sentences from each of the four datasets. However, only sentence pairs from Europarl was used for creating the training data for adaptive fine-tuning as Europarl's translation quality was the highest of the four datasets prepared; from the Europarl dataset, 2000 sentence pairs were used for the fine-tuning training data. For the German to English translations, an ensemble of three Transformer Big models are trained on the same data with identical parameters, but different initializers. They had 6 encoding and decoding layers, an rnn size of 512, a hidden transformer feed-forward layer of size 2,048, adam optimizer, and a decaying learning rate using noam decay method. The data was trained over the training data over one epoch with batche sizes of 4,096 and trained over the fine tuning training set over one epoch as well. Afterwards, the output translations are detokeized with the prior SentencePiece models and post-processed to fix any erroneous punctuation and replace previously removed tokens with a post-processing python script before a BLEU score is finally calculated with sacre bleu (Post, 2018). For comparison, one of the models in the ensemble serves as a baseline, but without any training runs over the fine tuning training data, but it does benefit from the same post-processing as the ensemble.

4 Results

After post processing the baseline model's translations of the testing set, SacreBleu calculated a BLEU score of 25.3, which is unimpressive, but not too surprising given the small amount of training data, single epoch of training, and lack of ensembling. However, the ensemble of three models, one of which was the baseline model after adaptive fine tuning, managed to score significantly lower with a BLEU score of 13.1. Such a drastic decrease in score after employing techniqes that are generally considered to bolster performance suggests that there is likely a larger problem in the workflow pipeline, but it is not immediately obvious what that would be.

5 Discussion

There are several areas in which the translation system could have been improved. The first was the very small amount of training data: 40,000 sentence pairs is not a lot for reaching BLEU scores comparable to teams at the WMT that would have multiple millions of sentence pairs on which to train their models. However, the makeup of the 40,000 sentence pairs is also a factor in the lower scores. Considering that the size of all of the original parallel corpora easily eclipsed 40,000 sentence pairs, it makes little sense to use sentence pairs from ParaCrawl or CommonCrawl, which are widely recognized as having lesser quality of training material, and it also makes little sense especially to use backtranslated data from a pretrained model, when there are still many sentence pairs remaining to draw from Europarl. Besides the size of the training data, the number of epochs for training the models was also hugely undersized. In order to get a better score, the models should have been trained on tens of epochs at least. Even the adaptive fine tuning, which generally sees a convergence in the gradient after very few epochs relative to the rest of the training steps, could have likely benefitted from another few epochs.

Additionally, the majority of ensembles I came across in the literatures would have 4-6 models in the ensemble and sometimes even more, so ensembling only three models for this translation system is not an optimal decision.

Another area for improvement is the heuristics by which robust text was substituted for tokens in the preprocessing of data and then reinserted in the post processing. While this does, ideally, ensure that the robust text is unadultered in the translation process, the post processing script simply reinserts the text back into the translation sentence in the same order that it extracted them; there is no way to tell which token belonged to which robust text. This then relies on the tokens being in the same order on the target side as they were on the source side despite the fact that an admitted challenge of translating German into English is the possible reordering of words in a sentence. However, even all these recognized issues do not explain how the translations of an ensemble of three models with fine tuning managed to achieve almost half the BLEU score of a single baseline model; this suggests that while addressing the issues mentioned above can boost the performance of this system by quite a bit, there is likely a far more critical issue with the workflow pipeline that needs to be fixed first in order to see considerable improvements and results.

6 Conclusion

The goal of this paper was to create an end-to-end translation system for German-to-English using a handful of state-of-the-art techniques in machine translation. To accomplish that, I had focused on preparing training data through careful data cleaning and filtering as well as back translation for data augmentation. I also attempted to create an ensemble of three fine-tuned models to follow the ubiquitous practice of ensembling and fine tuning for machine translation. Finally, I had trivial post processing to provide a small boost to model's translation quality. While optimistic about the results, the translation system's end performance does not come close to meeting expectations.

References

Rachel Bawden, Nikolay Bogoychev, Ulrich Germann, Roman Grundkiewicz, Faheem Kirefu, Antonio Valerio Miceli Barone, and Alexandra Birch. 2019. The University of Edinburgh's submissions to the WMT19 news translation task. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 103–115, Florence, Italy. Association for Computational Linguistics.

Alexandre Bérard, Ioan Calapodescu, and Claude Roux. 2019. Naver labs europe's systems for the WMT19 machine translation robustness task. *CoRR*, abs/1907.06488.

Ergun Biçici. 2019. Machine translation with parfda, Moses, kenlm, nplm, and PRO. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 122–128, Florence, Italy. Association for Computational Linguistics.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, André Martins, Christof Monz, Matteo Negri, Aurélie Névéol, Mariana Neves, Matt Post, Marco Turchi, and Karin Verspoor, editors. 2019. *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*. Association for Computational Linguistics, Florence, Italy.

Javier Iranzo-Sánchez, Gonçal Garcés Díaz-Munío, Jorge Civera, and Alfons Juan. 2019. The MLLP-UPV supervised machine translation systems for WMT19 news translation task. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 218–224, Florence, Italy. Association for Computational Linguistics.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. 2017. Opennmt: Open-source toolkit for neural machine translation. In *Proc. ACL*.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. *CoRR*, abs/1804.10959.

Bei Li, Yinqiao Li, Chen Xu, Ye Lin, Jiqiang Liu, Hui Liu, Ziyang Wang, Yuhao Zhang, Nuo Xu, Zeyang Wang, Kai Feng, Hexuan Chen, Tengbo Liu, Yanyang Li, Qiang Wang, Tong Xiao, and Jingbo Zhu. 2019. The NiuTrans machine translation systems for WMT19. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 257–266, Florence, Italy. Association for Computational Linguistics.

Kelly Marchisio, Yash Kumar Lal, and Philipp Koehn. 2019. Johns Hopkins University submission for WMT news translation task. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 287–293, Florence, Italy. Association for Computational Linguistics.

Alexander Molchanov. 2019. PROMT systems for WMT 2019 shared translation task. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 302–307, Florence, Italy. Association for Computational Linguistics.

Soichiro Murakami, Makoto Morishita, Tsutomu Hirao, and Masaaki Nagata. 2019. Ntt's machine translation systems for WMT19 robustness task. *CoRR*, abs/1907.03927.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook FAIR's WMT19 news translation task submission. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 314–319, Florence, Italy. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

Jan Rosendahl, Christian Herold, Yunsu Kim, Miguel Graça, Weiyue Wang, Parnia Bahar, Yingbo Gao, and Hermann Ney. 2019. The RWTH Aachen University machine translation systems for WMT 2019. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 349–355, Florence, Italy. Association for Computational Linguistics.

Felix Stahlberg, Danielle Saunders, Adrià de Gispert, and Bill Byrne. 2019. CUED@WMT19:EWC&LMs. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 364–373, Florence, Italy. Association for Computational Linguistics.

Meng Sun, Bojian Jiang, Hao Xiong, Zhongjun He, Hua Wu, and Haifeng Wang. 2019. Baidu neural machine translation systems for WMT19. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 374–381, Florence, Italy. Association for Computational Linguistics.

Yingce Xia, Xu Tan, Fei Tian, Fei Gao, Di He, Weicong Chen, Yang Fan, Linyuan Gong, Yichong Leng, Renqian Luo, Yiren Wang, Lijun Wu, Jinhua Zhu, Tao Qin, and Tie-Yan Liu. 2019. Microsoft Research Asia's systems for WMT19. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 424–433, Florence, Italy. Association for Computational Linguistics.

Renjie Zheng, Hairong Liu, Mingbo Ma, Baigong Zheng, and Liang Huang. 2019. Robust machine translation with domain sensitive pseudo-sources: Baidu-OSU WMT19 MT robustness shared task system report. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 559–564, Florence, Italy. Association for Computational Linguistics.