Translated - Machine Learning Assessment

Christian Brignone

Exercise 2 - Machine Translation

Suppose you work for a company that offers Machine Translation to its customers. You have access to a large amount of translation data (i.e., parallel files containing a sentence in a source language and the corresponding translation in the target language). Some of this data comes from public datasets, others have been produced by professional translators working for your own company.

Question 1 What kind of model would you use to implement a Machine Translation system? Describe its main features.

Answer 1 The kind of model that I would implement would certainly be a Transformer-based encoder-decoder architecture, in fact, since its introduction in the paper "Attention Is All You Need" by Vaswani et al. in 2017, the Transformer has become the dominant architecture for state-of-the-art Machine Translation systems. Among the many advantages brought by the transformer, one of its key features is the ability to scale to much larger datasets than previous Machine Translation models, and since we have a large amount of translation data at our disposal, this can lead us to significant improvements in performance.

Neural Machine Translation (NMT) models rely heavily not only on the quantity but also on the quality of training data. Therefore, the possibility of having access to high-quality data produced by professional translators can be incredibly advantageous. This is essential for building robust and accurate NMT models that can generalize well, produce high-quality translations, and be effective in specific domains.

In the hypothesis that our Machine Translation system will be used as a Computer-assisted Translation (CAT) tool by professional linguists, what could be a game-changer, would be the possibility of annotating and employing the post-editing corrections made by the professionals. This would allow for building an adaptive architecture that could use this information to customize the NMT model on the user style, just like Translated does with ModernMT.

The adaptive neural network would be fine-tuned at run-time on a subset of our large database, specifically selected in order to optimize the model on the domain, style and context of the document. This would certainly have a cost in terms of efficiency since it will require additional steps to adapt the network to the specific case but there will be significant improvements in terms of the quality of translation.

The ability to constantly learn from the new data provided by post-edits and adapt to customers' preferences would also avoid the need for training several models for domain-specific translations and to periodically re-train from scratch each model to update them.

Question 2 Imagine having to manage customers with different needs: some need the highest quality, by compromising on inference speed. Others need to scale the inference over a huge amount of data, and have time constraints (i.e., they want a translation obtained in a few milliseconds), while maintaining good quality. How would you handle the Machine Translation models in this scenario?

Answer 2 In the case in which I would have to manage customers with different needs, I would leverage the versatility of our model to adapt it to the scenario. In fact, for customers needing the highest quality, I would employ our adaptive neural network allowing the model to continuously fine-tune itself on specific data, thus providing the best possible translation.

While for customers interested in obtaining translation in a few milliseconds, we can simply use our model as it is, without fine-tuning it at every iteration. In this way, the neural network will be much faster by compromising on the quality increase given by the continuous adaptation to the specific context. Since we would employ a transformer-based architecture trained on a very large dataset of high-quality data, we will still maintain a good-quality translation.

Obviously, if the inference time requirements are very tight, and we have to speed up inference even more, we may have to build a lighter version of the model to meet them.

Question 3 How would you implement a system to monitor the quality of a Machine Translation system currently in production?

Answer 3 To monitor the quality of a Machine Translation system already in production, other than the usual combination of automatic evaluation metrics like BLEU, COMET and TER, I would mainly rely on the Time to Edit (TTE), which is the average time per word employed by a professional translator to correct the output of our model. In fact, the time needed for a professional to correct the automatic translation is directly linked to the translation quality: the higher the quality the lower the time needed to post-edit. Furthermore, TTE is able to grasp all the nuances of the problem, since it can be seen as a measure of the cognitive effort made by the translator. In fact, in many situations, even if only a single word is corrected by the professional, if the correction of that single word would require a very long time and effort for him, the suggestion given by the model should be not considered good. This kind of phenomenon is not captured by standard automatic evaluation metrics which instead may evaluate it as relatively good.

Another important metric to measure the quality of the models currently in production is the Errors per Thousand (EPT) words. This can be calculated after the reviews by professional translators and gives the average number of linguistic errors contained in a thousand words. EPT is strictly related to the quality of the suggestions, the lower the EPT the higher the quality.

A fundamental aspect of TTE and EPT is that, differently from other human evaluation criteria, both metrics can be automatically measured while translators are doing their job using CAT tools thus not requiring additional time and money to directly evaluate the quality of the suggestions.