Exercise 2 - Machine Translation

Gioia Mancini

April 10, 2023

Introduction

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

1 Question 1

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

1.1 Answer 1

Machine Translation (MT) is a task whose aim is to translate natural language sentences from a source language to a target language. Early methods relied on rule-based machine translation or Statistical Machine Translation (SMT), recently replaced by the new paradigm of Neural Machine Translation (NMT), which employs deep neural networks to translate languages [1].

A specific solution I would employ to design a high-quality MT system, given the availability of a large amount of parallel corpora as training data, is a Transformer-based encoder-decoder neural network with attention mechanism. In fact, from the recent milestone paper Attention is All You Need [2], Transformer architectures became pervasive in the development of new NMT systems. This architecture uses attention mechanisms to model dependencies between source and target sentences, allowing the model to capture long-term dependencies and produce accurate translations. This kind of model has the ability to handle variable-length input and output sequences, bidirectional encoding, and multiple source sentences simultaneously.

State-of-the-art NMT are powerful tools for giving a fluent and high-quality translation in almost any kind of area, often resulting in outstanding performance. In particular, the design of a multilingual MT model should be considered, with training performed with parallel corpora that contain sentence pairs in multiple languages. These sentence pairs can be aligned at the sentence level so that the corresponding sentences in different languages are paired together. A shared encoder-decoder architecture allows the model to transfer knowledge between different languages, which can be particularly beneficial for low-resource languages that may not have enough data to train a high-quality MT model independently.

Furthermore, I would take into consideration a context-aware NMT model, which could be a good strategy to improve the quality of translation. In fact, using contextual information, e.g. surrounding sentences, the model would be able to generate more appropriate sentences with respect to the context, topic and style of the document to translate. This could be achieved by using techniques such as contextual embeddings given by a BERT model to inform the NMT model about the surrounding text context. Another example is conditional encoding or adaptive inference techniques such as domain or style adaptation and translation quality adaptation.

From this, a natural extension is to employ adaptive NMT, which is based on continually adapting the model to new data and changing conditions. A continual learning approach could be suitable, since the large number of post-edits provided by linguists which are at our disposal [3]. The human-in-the-loop paradigm embraced by Translated is a perfect strategy, in fact, the quality of the MT system can be constantly improved by adapting the NMT model at run-time, by means of the so-called translation memory employed for the ModernMT and Matecat tools [4]. This allows using a single model while continually adapting the machine to the translator and vice-versa, without having to re-train from scratch different models for every domain or specific context, which would be impractical for computational power, time and economic resources. Other effective strategies could also be employing reinforcement learning on post-edits information. In addition, this interaction between the translator and the MT system should happen in real-time, thus the overall performance should be combined with high speed of computation.

2 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?

2.1 Answer 2

Assuming that the company's goal is to reach a broad range of customers, we would be in presence of different customers' needs, ranging from basic use cases to more demanding ones. In this case, the most important thing would certainly be to provide a different range of products or services in order to meet each kind of customer's need. As I discussed in the previous section, it is important to carefully use resources in terms of computational effort, time efficiency and budget requirements. Therefore, it would be impractical to deploy several different MT systems for every purpose. One idea could be to develop a few NMT models, differing in architecture in terms of size, number of layers and inference speed.

In the case of the highest quality requirement and no particular need for inference, I would offer a system based on a big transformer with multiple attention heads and layers, trained on a large amount of data. Additionally, especially when the application is very industry-specific, such as medical/healthcare or finance, the model could be fine-tuned through an adaptive strategy based on context, post-edits and positive/negative examples generated by linguistics' run-time corrections. This would result in a high level of translation accuracy, perfectly adapted to the customer's needs and to the desired areas of interest.

On the other hand, for customers which require fast inference time or have to deal with time-sensitive or real-time applications, the design of the model could be different. In particular, I would design a lightweight transformer-based model with fewer attention heads and layers, in order to meet the efficiency and speed requirements while maintaining a reasonable level of quality. Moreover, since the adaptive strategy based on fine-tuning would require additional computational power and an increase in inference time, it could be avoided in this case, thus allowing a good translation even in a few milliseconds.

3 Question 3

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

3.1 Answer 3

One of the most important things in MT systems is the measurement of the quality of an obtained translation. Commonly used approaches are automatic metrics and human evaluation. Automatic metrics are essential tools for evaluating the quality of MT at the design and development level. As many research examples show, metrics such as BLEU, Lepor, TER, COMET and BertScore scores are employed to perform MT quality assessment. Automated measurement metrics are essential in data-driven MT systems, especially in development phases where quick feedback is needed to check the progress of the system.

The BLEU (Bilingual Evaluation Understudy) score is the most widely employed, however, it measures string similarity often relying on only one correct translation reference, thus its use could be inappropriate since in MT multiple translations can be technically correct. Moreover, since adaptive NMT systems evolve rapidly to different use cases, contexts and translators' competence, it is important to continuously measure and monitor the quality of the system during production. In this case, static automated metrics such as BLEU-based ones, cannot be employed.

I would use metrics to measure the post-edit effort, such as the edit distance, which can be used to measure the similarity between the machine-generated translation and the post-edited translation, giving a hint on the minimum number of operations needed to transform one text into another. However, this metric has strong limitations, since it considers only the number of changes made, and not the time spent working.

Instead, Time to Edit (TTE) is a more suitable metric for measuring long-term MT quality. It represents the total time spent by a human translator in post-editing a text, divided by the number of words of which that text is made. Translated currently uses this metric to measure the cognitive effort required to correct a translation, together with the so-called Errors per Thousand (EPT) words, measuring the linguistic errors done in 1000 words. As the TTE continues to improve, the EPT rate continues to decrease, leading to the generation of more and more accurate translation content.

References

[1] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[3] M. Turchi, M. Negri, M. Farajian, and M. Federico, "Continuous learning from human post-edits for neural machine translation," *The Prague Bulletin of Mathematical Linguistics*, vol. 108, no. 1, pp. 233–244, 2017.

[4] N. Bertoldi, R. Cattoni, M. Cettolo, M. A. Farajian, M. Federico, C. Davide, M. Luca, R. Andrea, T. Marco, G. Ulrich *et al.*, "Mmt: New open source mt for the translation industry," in *Proceedings of The 20th Annual Conference of the European Association for Machine Translation (EAMT)*, 2017, pp. 86–91.