

# Adversarial Cleaning of Translation Memory

Tientso Ning

2019, December 17

## Headline

Translation Memory is a computer-assistant tool to assist professional translators in their work.

## Summary

Due to the nature of Translation Memory software, lots of erroneous or incomplete entries need to be cleaned from time to time in order to maintain efficiency.

## The Purpose

We want to apply adversarial training to the process of cleaning translation memory, allowing for efficient unsupervised cleaning of duplicate, incomplete, and/or erroneous entries.

## The Problem

Translation Memory is considered one of the best computer-assistance tools for professional translators, and its ability to leverage the contributions of many industry professionals cannot be overlooked. Yet, translators are only able to work more efficiently and produce better translations with good, clean, Translation Memory. We want to have a simple yet effective way of cleaning Translation Memory systems.

## The Set-up

Previous studies have looked at comparing entries to machine-translated outputs in order to detect potentially spurious entries Barbu (2015), as well as leveraging part-of-speech information directly onto the entries in question Zwahlen, Carnal, and Läubli (2016). However, since the likelihood of spurious entries increases with the number of entries (as a function of time), the data is present to leverage applying adversarial training in the format of a generator/discriminator model in order to learn to clean the translation memory.

## **Your Approach**

Generative Adversarial Nets (GAN) is a machine-learning concept attributed to Ian Goodfellow Goodfellow et al. (2014), whereas two networks, the generator and discriminator compete to maximize and minimize the other's errors, respectively. In this fashion, the generator learns to produce an output similar to the given training distribution, while the discriminator learns instead to classify outputs to their given distribution. A known dataset is often used to train the generator. Applied in the case of cleaning translation memory, the generator is concerned with producing outputs similar to the distribution of entries that are proper translation memories, while the discriminator learns classification of which distribution, proper or erroneous, a given translation memory entry is from. Training data can be leveraged directly from the entries stored in the translation memory, allowing the learning process to be semi-supervised. This means that we can either do iterative unsupervised training on a translation memory dataset that is considered not yet inefficient or mostly efficient, having the generator generate entries similar to the distribution of the entire memory, or have labelled bad entries that the discriminator learns to identify. Both approaches will give us the ability to clean translation memory.

## **What you've done**

Evaluation of the effectiveness of the approach will be done in two parts: the accuracy of the model in cleaning erroneous translation memory entries, and the quality of the entries produced by the generator model. The reasoning for obtaining the former is quite clear, with the goal of the approach being to accurately remove erroneous entries from translation memory, the accuracy can be calculated based on how many of the entries manually labelled erroneous are correctly tagged for removal by the model. The reasoning for obtaining the latter is more concerned with additional information the model can provide in making translation memory even more efficient and usable for professional translators. Since the generator is tasked with producing outputs similar to the distribution of the entries in translation memory that is considered proper translations, the quality of these outputs can also be examined, for the potential to assist in classification or directly in translation tasks. These generator output pseudo-translations can be evaluated using a standard BLEU score, and can also be compared to the erroneous entries in a supervised manner in attempts to replace those erroneous entries with these output entries from the generator instead.

## **Conclusion/Further work**

Clean and accurate translation memory entries allows professional translators to leverage one of the core advantages of translation memory, which is the collective nature of the database. This allows translators to work quickly without having to re-translate already seen segments, as well as give translators a sort of redundancy to reduce translation errors. Over time, these databases have the tendency to be clogged with duplicate, incomplete, and/or erroneous entries that hinder the effectiveness of the program. Having a semi-supervised method in removing these hindering entries while providing additional functions to assist translations could have a positive impact in maintaining a good work-flow. Future considerations can look into other methods in detecting or improving translation entries that require less

overhead, as machine learning trends toward complexity. Additionally, there is also room to evaluate whether or not machine translation or machine produced translations could assist in improving the translation work-flow.

## References

- Barbu, Eduard. 2015. “Spotting False Translation Segments in Translation Memories,” September. Hissar, Bulgaria: Association for Computational Linguistics, 9–16. <https://www.aclweb.org/anthology/W15-5202>.
- Goodfellow, Ian J., Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. “Generative Adversarial Networks.”
- Zwahlen, Alena, Olivier Carnal, and Samuel Lüubli. 2016. “Automatic Tm Cleaning Through Mt and Pos Tagging: Autodesk’s Submission to the Nlp4tm 2016 Shared Task.”