Université de Montréal

Open Source Quality Control Tool for Translation Memory

par

Bhardwaj, Shivendra

Département d'informatique et de recherche opérationnelle
Faculté des arts et des sciences

Mémoire présenté en vue de l'obtention du grade de
Maître ès sciences (M.Sc.)
en Informatique

August 5, 2020

Université de Montréal

Faculté des études supérieures et postdoctorales

Ce mémoire intitulé

Open Source Quality Control Tool for Translation Memory

présenté par

Bhardwaj, Shivendra

a été évalué par un jury composé des personnes suivantes :

Jian-Yun Nie

(président-rapporteur)

Philippe Langlais

(directeur de recherche)

Emma Frejinger

(membre du jury)

Résumé

La mémoire de traduction (MT) joue un rôle décisif lors de la traduction et constitue une base de données idéale pour la plupart des professionnels de la langue. Cependant, une MT est très sujète au bruit et, en outre, il n'y a pas de source spécifique. Des efforts importants ont été déployés pour nettoyer des MT, en particulier pour former un meilleur système de traduction automatique. Dans cette thèse, nous essayons également de nettoyer la MT mais avec un objectif plus large : maintenir sa qualité globale et la rendre suffisament robuste pour un usage interne dans les institutions. Nous proposons un processus en deux étapes : d'abord nettoyer une MT institutionnelle (presque propre), c'est-à-dire éliminer le bruit, puis détecter les textes traduits à partir de systèmes neuronaux de traduction.

Pour la tâche d'élimination du bruit, nous proposons une architecture impliquant cinq approches basées sur l'heuristique, l'ingénierie fonctionnelle et l'apprentissage profond. Nous évaluons cette tâche à la fois par annotation manuelle et traduction automatique (TA). Nous signalons un gain notable de +1,08 score BLEU par rapport à un système de nettoyage état de l'art. Nous proposons également un outil Web qui annote automatiquement les traductions incorrectes, y compris mal alignées, pour les institutions afin de maintenir une MT sans erreur.

Les modèles neuronaux profonds ont considérablement amélioré les systèmes MT, et ces systèmes traduisent une immense quantité de texte chaque jour. Le matériel traduit par de tels systèmes finissent par peuplet les MT, et le stockage de ces unités de traduction dans TM n'est pas idéal. Nous proposons un module de détection sous deux conditions: une tâche bilingue et une monolingue (pour ce dernier cas, le classificateur ne regarde que la traduction, pas la phrase originale). Nous rapportons une précision moyenne d'environ 85 % en domaine et 75 % hors domaine dans le cas bilingue et 81 % en domaine et 63 % hors domaine pour le cas monolingue en utilisant des classificateurs d'apprentissage profond.

Mots-clés —Traduction automatique, nettoyage d'une mémoire de traduction, alignement de phrases, détection de traduction automatique, réseau neuronal profond, transformateur profond, ingénierie de traits, classification

Abstract

Translation Memory (TM) plays a decisive role during translation and is the go-to database for most language professionals. However, they are highly prone to noise, and additionally, there is no one specific source. There have been many significant efforts in cleaning the TM, especially for training a better Machine Translation system. In this thesis, we also try to clean the TM but with a broader goal of maintaining its overall quality and making it robust for internal use in institutions. We propose a two-step process, first clean an almost clean TM, i.e. noise removal and then detect texts translated from neural machine translation systems.

For the noise removal task, we propose an architecture involving five approaches based on heuristics, feature engineering, and deep-learning and evaluate this task by both manual annotation and Machine Translation (MT). We report a notable gain of +1.08 BLEU score over a state-of-the-art, off-the-shelf TM cleaning system. We also propose a web-based tool "OSTI: An Open-Source Translation-memory Instrument" that automatically annotates the incorrect translations (including misaligned) for the institutions to maintain an error-free TM.

Deep neural models tremendously improved MT systems, and these systems are translating an immense amount of text every day. The automatically translated text finds a way to TM, and storing these translation units in TM is not ideal. We propose a detection module under two settings: a monolingual task, in which the classifier only looks at the translation; and a bilingual task, in which the source text is also taken into consideration. We report a mean accuracy of around 85% in-domain and 75% out-of-domain for bilingual and 81% in-domain and 63% out-of-domain from monolingual tasks using deep-learning classifiers.

Keywords —Machine Translation, Cleaning Translation Memory, Sentence Alignment, Machine Translation Detection, Deep Neural Network, Deep Transformer, Feature Engineering, Classification Model

Contents

List of Tables

List of Figures

Liste des sigles et des abréviations

TM	Translation Memory
TU	Translation Unit
BT	Bureau Translation
MDTM	Multi-domain translation memory
SP	Sentence Pair
MT	Machine Translation
NMT	Neural Machine Translation
SMT	Stastical Machine Translation
KL divergence	Kullback–Leibler divergence
RF	Random Forest
SVM	Support Vector Machine

LSTM	Long Short Term Memory
LASER	Language-Agnostic SEntence Representations
MSS	Multilingual-Similarity Search
XLM	Cross-lingual Language Model Pretraining
ConvS2S	Convolutional Sequence to Sequence
GT	Google Translate
D	DeepL
EURO	Europarl
HANS	Canadian Hansard
NEWS	News Commentaries
CRAWL	Common Crawl
BPE	Binary-Pair Encoding
SRC	Source

TGT	Target
HUM	Human
CAT	Computer-assisted translation
HTML	Hypertext Markup Language
TMX	Translation Memory eXchange
NLTK	Natural Language Toolkit
RAM	Random Access Memory

Remerciements

I would first like to thank my thesis director Professor Philippe Langlais. The door to Prof. Langlais office was always open whenever I ran into a trouble spot or had a question about my research or writing. He consistently allowed these papers to be my work, but steered me in the right direction whenever he thought I needed it.

I would also like to thank the NRC-CNRC experts Michel Simard, Cyril Goutte and Gabriel Bernier-Colborne, who were involved in the proofreading of the three papers. Without their passionate participation and input, the papers could not have been successfully submitted to COLING'2020.

I would also like to acknowledge David Alfonso-Hermelo at RALI for his vital support during the entire process, and I am gratefully indebted to his extremely valuable work in the papers. I want to thank all the people at the RALI who helped me during my research through endless theoretical and practical discussions, especially Abbas, David, Fabrizio, Guillaume, Ilan, Khalil, Olivier and Vincent. Apart from work, I had too much fun with RALI members; particularly, the crazy board games always helped during my stressed times.

Finally, I must express my very profound gratitude to my parents, my girlfriend and my friends for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you.

Chapter 1

Introduction

1.1. Context

Data is the most valuable asset and we are collecting this for ages now. One such database is named Translation Memory (TM). It stores sentences, paragraphs or sentence-like units along with their translation (translation unit) in different languages, a.k.a parallel corpus. A TM plays a critical role in translation services and is the bread and butter for most language professionals. These TMs are systematically matched against their content for reuse and increasing consistency. The language professionals consider it their primary source of information and query it directly through their "concordance" functionality [Bundgaard and Christensen, 2019, Teixeira and O'Brien, 2017].

The classic advantage of storing good quality TM is that one never has to translate the same sentence again. The professional translators can maintain consistent word choice throughout and across documents, which is critical to achieving a high-quality translation. In this manner, they can save time and also maintain the overall quality. At last, the industries and institutions can save money, along with the fast translation, and less-experienced translators can be put to good use. More recently, TM contents have also become the primary source of fuel for domain- or client-specific machine translation systems.

Many institutions are collecting TMs for a long time, and, notably, most TMs contain noise, which reduces their reuse. This noise accumulation can occur from translation-specific issues such as missing or untranslated parts in target documents, the use of calques, improper wordings, wrong spelling, morphosyntax errors and many more. Another type of noise accumulates due to all the machineries (data flow pipelines) involved in the collection process of TM. Improper alignment and segmentation are common types of noise generated through these pipelines, which can render some pairs, and sometimes whole documents, unusable. Furthermore, while using a TM, these accumulated noise hinders the overall translation process. Also, the language professionals can no longer trust the resource which contains such noise. In the case of machine translation systems, however, the Statistical Machine Translation (SMT) systems were known to be resilient to noise [Goutte et al., 2012], the situation is quite different with Neural Machine Translation (NMT) systems [Khayrallah and Koehn, 2018].

This noise also increases with time and at a much faster rate than we imagined. The effect of "error propagation" is one of the main motivations for cleaning a TM regularly. The incorrect translation (noise discussed above) will be reused again and again for translating the same source text (or similar text), thereby perpetuating the error. It also has a severe effect on the quality of machine translation systems, since these TMs will be employed for training. With this, the requirement for automatic and efficient cleaning and storing error-free TM emerged.

1.2. Motivation

The Translation Bureau of Canada [1] is a federal institution which supports the Government of Canada in maintaining the two official languages (English, French) throughout the country. Over the years, the Translation Bureau has collected an enormous TM, with a high intrinsic value. The collected TM however, is prone to noise, and the Translation Bureau started questioning their TM quality and its collection process (complete pipeline). These circumstances motivated the need for methods for "cleaning" TMs. To articulate this problem, the Translation Bureau of Canada collaborated with Recherche Appliquée en Linguistique Informatique (RALI) lab at Université de Montréal and National Research Council Canada/Le Conseil National de Recherches Canada (NRC-CNRC) (Multilingual Text processing team [NRC]).

The Translation Bureau of Canada provided us with an extensive TM (139M sentence pairs), which is multi-domain (200+ different domains). We called it *Multi-domain translation memory* (MDTM). It is the result of collaborative works done by many professional translators and language professionals at the Translation Bureau. We have conducted the experiments with a large sample extracted from MDMT.

Many research works proposed are tackling noise removal for rather noisy TM, such as, for instance, ones collected from the web (discussed in Chapter 2). In this thesis, we are more concerned with cleaning a TM, which is overall of good quality since produced by professional translators. This thesis articulates this problem from two directions: pure cleaning, that is by detecting noise and second being, detecting machine-translated sentences in the TM.

[1] `https://www.tpsgc-pwgsc.gc.ca/bt-tb/index-eng.html`

Apart from the noise described above, the presence of translations produced from automatic machine translation systems in TM is also a known issue (Section 2.3). The translation produced from such automatic systems sometimes lacks coherence and context. Therefore, to have pure context-based and well-formulated translations stored in TM, it is common to filter out such texts.

Additionally, the quality of automatic translation systems (SMT & NMT) will not improve, considering it will be trained on data produced from such systems rather than professional translators. Also, detecting machine-translated sentences in TM can help organizations find out if translator strongly relied on automatic systems rather than utilizing the language knowledge, context and domain understanding.

We proposed an "Open Source Quality Control Tool for Translation Memory", a complete package that does the noise removal and maintains the quality of a large institutional TM. This tool was developed from scratch at RALI with computational support (Linux environment and GPUs) from NRC-CNRC. In order to design this tool, we propose and study a three-step solution which is presented into three different articles, each corresponding to one chapter in this thesis:

- Cleaning a(n almost) Clean Institutional Translation Memory.
- Human or Neural Translation?
- OSTI: An Open-Source Translation-memory Instrument.

Authors for the above three articles are, Cyril Goutte [2], David Alfonso-Hermelo [3], Gabriel Bernier-Colborne[2], Michel Simard[2], Philippe Langlais[3], Shivendra Bhardwaj[3]. We have submitted the three proposed articles to COLING2020 [4] on July 1st, 2020 and will receive an acceptance notification on October 1st, 2020. While writing, we had many constructive discussions with the team at NRC-CNRC (Michel Simard, Cyril Goutte and Gabriel Bernier-Colborne), and the team also proofread all three articles. The contribution of each author is mentioned at the beginning of each chapter.

1.3. Realisations

This master thesis reports in detail the concepts, architectures and the implementation of the overall package, "Open Source Quality Control Tool for Translation Memory". This package is sub-divided into two broader sections, first, filtering a mostly clean institutional translation memory and second on neural machine translation detection task. In the upcoming sections, I will introduce the three articles on which this thesis is built.

[2] Researchers from NRC-CNRC

[3] Researchers from RALI

[4] https://coling2020.org/

1.3.1. First Step

The first step also our first paper is the soul of this architecture. We have already discussed how important it is to have a clean TM, and with a regular cleaning, we can mitigate the risk of the "error propagation" effect. In order to tackle this problem, we compare five approaches involving solutions based on heuristics, feature engineering, and deep learning. The evaluation was done through both manual annotation (by a language professional) and State-Of-The-Art (SOTA) machine translation systems. The proposed method proclaims a noticeable gain over a SOTA off-the-shelf Translation Memory cleaning system. The complete architecture and results are discussed in detail in Chapter 3 in the form of a paper named "Cleaning a(n almost) Clean Institutional Translation Memory" which puts forward up-to-the-minute Artificial Intelligence (AI) techniques and approaches.

From here, we will try to capture the birds-eye view of the approach and steps we followed in a sequential manner. In this work, a sample from MDTM was examined (Source: English, translation: French) with carefully accommodating comparable domain distribution, dubbed as MT-TRAIN in Chapter 3. Our first step was to understand the data and determine the different kinds of noise present through a manual annotation of a small, systematically obtained sample from MDTM. A language expert did this rigorous process at RALI, David Alfonso-Hermelo (in short, David). With the learning from annotation and literature, David designed a set of customized heuristics for detecting noisy and clean translation.

We applied these heuristics on the whole MDTM (139M SP), which provided us with the 17.7M sentence pairs classified as "good" (being in the translation relation with some possibility of noise), and 1.2M sentence pairs as "bad" (noisy/not in translation relation). The heuristics are focused entirely on getting the highest precision possible but it did had low recall, i.e., there was some level of misclassification in each set. Also, most of the SPs were classified as silence, i.e., the heuristics could not decide with certainty if the SP was "good" OR "bad". This step considered hard threshold functions (each heuristic being functions with hard boundaries), and there are appreciably finer errors that we could not trustworthy extract with such fixed bounds. Therefore, this formulates the need for a robust method to best capture the more subtle features indicating translation relation and noise.

The previous step scratched the tip of an iceberg; the finer noise is much more complex and hard to detect by heuristics, which are basically a set of linear functions. The original literature of heuristics [Barbu, 2015], on which we designed our maximum heuristics, did showcase good accuracy albert showing a high false-negative rate. Barbu [2015] did confirm that: it is difficult to clean an almost clean corpus using just heuristics, since it is prone to misclassification errors. The heuristics were designed from a small sample, although we extracted this sample systematically, it still does not represent the complete noise umbrella we want to target.

To tackle this, I (co-author) proposed two mainstream deep-learning-based approaches, supervised- and unsupervised-learning methodologies. In order to extract more complex features deep neural networks was utilized. For evaluation, I trained two SOTA machine translation systems XLM [Lample and Conneau, 2019] and ConvS2S [Gehring et al., 2017]. Trained two NMT models from scratch on multiple data sets including, MT-TRAIN (sampled data from MDMT) and clean (heuristics-, feature-, deep-based). We have also measured the BLEU score (measured with two machine translation systems) at each step from MT-TRAIN to deep-based cleaning.

The motivation for the unsupervised approach is to skip the human intervention, which might be biased and definitely costly. We employed LASER [Artetxe and Schwenk, 2019a], a module capable of pointing out non-parallel sentence pairs without any input labels (completely in a unsupervised way). LASER was directly applied to clean MT-TRAIN; this way, we escaped the human intervention from the entire TM cleaning pipeline. LASER showed a noticeable gain in BLEU score, comparable to the best results obtained by supervised learning approaches.

In a supervised learning approach, I proposed an LSTM [Hochreiter and Schmidhuber, 1997b] based architecture which was inspired from Grégoire and Langlais [2018]. Our LSTM based architecture consumed the heuristic output as two-class data (as mentioned above, 1.2M “bad” and 17.7M as “good”), one with inconsistencies and another a supposedly clean translation. In addition to that, I have also designed artificial noise (fine-grained) from the learning of Grégoire and Langlais [2018] and Bernier-Colborne and Lo [2019]. Along with this, David explored feature-based classifiers, such as support vector machines (SVM) [Hearst et al., 1998] and random forests (RF) [Breiman, 2001].

We noted a continuous gain in BLEU score at each cleaning step and recorded a notable gain from a deep-learning-based approach. Both LASER and LSTM performed well within a close margin. BLEU scores is not the only factor on which we base our results, but we also thoroughly analyzed the cleaned data from each method. Thanks to Professor Philippe Langlais, who helped me side-by-side with his deep language skill in writing the analysis section in Chapter 3. To get the fine-rate cleaned data, we considered the intersection of the two best models (LSTM and LASER) dubbed as ∩ALL. This ∩ALL data showcased a marginal gain in BLEU over the previous best score.

1.3.2. Second Step

In this section, we will discuss our motivation behind detecting machine translated texts and provide an overview of our proposed solution. The work is discussed in detail in Chapter 4 as a paper named, “Human or Neural Translation?”.

With the advancement in the machine translation systems (discussed in Chapter 2.1.2), machine translation is now as close as a mouse click. Popular and easily accessible translation engines like Google Translate [5], DeepL [6], Bing Translate [7] and many other in-house trained NMT models are translating enormous amounts of data on a daily basis. These translation engines are available to all, and it gets used by professional translators as well to ease their job.

Moreover, some institutions require data to be translated by professional (human) translators to maintain the quality. These institutions do expect professional translators to use the internally available TMs for reference and a minimal use of machine translation systems. Institutional data such as court-room decisions, medical reports, defence agreements, research papers, bilateral contracts between countries; require specific domain knowledge and related references for the translation. If professional translators use automatic systems to translate; these institutions may suffer from the issues like data quality, and integrity.

This issue was not there during Statistical Machine Translation (discussed in Chapter 2.1.1) since the translation was not as good as the current SOTA NMT models. Although these NMT models are most desirable, there are still producing some flaws during translation. The translation quality falls for long sentences, complex sentences with multiple phrases and in case of inter-sentence dependencies. Those problems are well-known, and current machine translation engineers are still trying to solve them. Despite all those flaws, it is tough for humans to determine whether a machine or a professional translator did the translation. Therefore the question emerged, whether we could develop a tool to distinguish such sentences or sentence pairs.

While writing this, the stated problem is still an active area of research but, the community has not focused on solving this problem for real-life data produced by SOTA NMT systems. We propose a tool to detect machine-translated sentences by exploiting SOTA Transformer [Vaswani et al., 2017] based on language models and machine translation systems. The tool consists of two settings: a monolingual one, where we only feed the translation to learn and predict, and a bilingual one, where we pass the source sentence as an additional input. To build this tool, I have trained 18

[5] https://translate.google.ca/
[6] https://deepl.com/translator
[7] https://www.bing.com/translator

different feature- and deep-learning-based detection models, and two in-house NMT models from scratch. For out-of-domain experiments, I have employed publicly available translation engines, Google Translate and DeepL. The entire module was designed from scratch and developed at RALI with the computational help from NRC-CNRC (GPUs and Linux environments).

The development of this module is divided into multiple steps, the first being, collecting "Human Translated" (HT) corpus of feasible size to train NMT models. The second and the most crucial step is translating a big chunk of HT data (apart from data used for training the NMT models), which acts as the "Machine Translated" (MT) data for the experiment. Finally, we formulate a SOTA detection module that can efficiently find out a machine-translated text (or a sentence pair) in a TM.

We employed the $\cap$ALL data from Chapter 3 as a HT corpus to train two SOTA NMT models, XLM [Lample and Conneau, 2019] and Scaling NMT [Ott et al., 2018b]. Next, we translated a separate chunk of HT corpus using the two trained NMT models. This step consolidates the two MT corpora, one from each of the two NMT models. With this, we constructed the base for the research work by creating parallel HT and MT data sets.

The detection module of the tool was developed by exploiting transfer learning, deep learning, and feature-based classifiers. CamemBERT [Martin et al., 2019] , XLMRoBERTa [Ruder et al., 2019], FlauBERT [Le et al., 2020], XLM [Lample and Conneau, 2019], mBERT [Devlin et al., 2018a] pre-trained models were incorporated as the transfer learning approaches. Under the deep learning umbrella, LSTM [Hochreiter and Schmidhuber, 1997b] and LASER [Artetxe and Schwenk, 2019a] were utilized, and subsequently feature-based methods *N*-gram [Cavnar and Trenkle, 1994], KenLM [Heafield, 2011] and T-MOP [Jalili Sabet et al., 2016b] were exploited.

Along with the in-domain data ($\cap$ ALL), we tested this module for out-of-domain data sets like Europarl, the parliament debates of Canada, the News Commentaries, and the Common Crawl corpus. We translated a sample of these data sets thanks to Google and DeepL to show that our method can generalize to other domains and translation engines. Special thanks to Professor Philippe Langlais, who provided his language understanding and profound knowledge in the translation domain to formulate the analysis section in Chapter 4. We have discussed in detail about different aspects of translations on which our models base their decisions. Professor Philippe Langlais proposed this problem statement, and I designed and conducted all the experiments. In the writing part, I, Professor Philippe Langlais and David contributed.

1.3.3. Third Step

Maintenance of TMs still tends to be a manual process in most cases, and allocating human resources to clean noisy TM is a tedious and costly affair. We propose an open-source and freely available web-based tool (in Chapter 5) to process and visualize a language pair documents (parallel corpora) into an automatically labelled (noisy or not) translation units. These assigned labels will internally help the Translation Bureau to control and monitor the noise. This paper explores the best methodologies discussed in the first paper (Chapter 3).

The tool consists of four-step: First, we segment each text into sentences; then, we align the TM at the sentence level. Third, aligned sentence pairs are labelled as either good or noisy. The final step labels it into six classes for human readability on an HTML interface that allows us to export the automatically or manually selected SPs into a TMX file.

The tool was presented as a Demo paper "OSTI: An Open-Source Translation-memory Instrument" in Chapter 5 and submitted at COLING2020 conceptualized as web-based Bitext Alignment Tool. This quality control tool was developed at RALI for the institutions to maintain an error-free TM. This paper includes the best approach proposed in Chapter 3, LASER, as a detection and alignment module. David, me and Professor Philippe Langlais worked on the analysis and the writing part.

1.4. Thesis Chapters details

The thesis is distributed across five chapters including this one. Chapter 2 discusses the state-of-the-art produced over the years in the domain of TM cleaning and machine translation detection. In Chapters 3, 4 and 5, we will discuss the three articles in-details. The last Chapter 6 concludes the thesis and the results.

Chapter 2

Literature Survey

This chapter first discusses the literature related to the paper “Cleaning a(n almost) Clean Institutional Translation Memory” and then talks about the State-Of-The-Art (SOTA) for the paper “Human or Neural Translation?”.

2.1. Machine Translation

We live in a world where language and distance are not a barrier, and we collaborate with people around the globe regularly. Developments in the communication sector (like telephones and video conferencing) diminished the distance barrier. To overcome the language barrier (arguably 6500 spoken languages in the world), the concept of a professional translator (expert in two or more languages) pitched in. It built a bridge for the language gap.

A professional translator understands the text or speech in one language and delivers the meaning to another. It is not a straightforward job; with the correct translation, they must maintain the context during the translation with precision. Professional translators are helping the world communicate and collaborate on almost all platforms such as education, medical, defence, and many more. In the past few decades, the difficulty for a translator has reduced slightly with the influx of machine translation systems (Statistical and Neural Machine Translation). In order to train such translation systems, Translation Memory (TM) is the fuel.

From here, we will discuss the evolution of Statistical and Neural Machine Translation to the current state-of-the-art approaches and understand how error-free TM is crucial during that process.

2.1.1. Statistical Machine Translation and IBM Models

The research in the area of statistical speech recognition systems [Bahl et al., 1983] was very popular during the ’80s, which researchers later studied in combination with Natural Language Processing (NLP) [Baker, 1979, Garside et al., 1989, Sampson, 1986, Sharman et al., 1990]. These studies encouraged the research on Statistical Machine Translation (SMT) [Brown et al., 1990]. One of the initial works, like that of Brown et al. [1990], evaluates the joint probability of the source (S) and target/translation sentence (T) using the Bayes' theorem. The authors of Brown et al. [1990] considered only the translation of individual sentences. Their method assigns a score to each sentence pair based on the probability that a translator will translate the sentence T when given with the source sentence S. During the translation step, it evaluates the maximum probability of a source sentence S given the target sentence T, and this determines the best translation.

To train such a SMT model, TM is required (fuel), and extracting a TM was initially introduced by Brown et al. [1991] and Gale and Church [1993b]. The research work by Brown et al. [1991] focuses on the number of words, while Gale and Church [1993b] is based on the number of characters sentence contains. These two approaches also inspired the IBM word-alignment models [Brown et al., 1993b], one of the popular methodologies for statistical machine translation. IBM models dominated the translation realm until the maturity of phrase-based SMT finally superseded by the Neural Machine Translation (NMT).

Brown et al. [1993b] proposed five translation models, also known as IBM Models. The IBM models are purely word-based models and capable of capturing local context in the sentence. IBM Model 1 trains a simple word alignment model that uses the Expectation-Maximization (EM) algorithm [1] [Baum, 1972], as is the case with the other IBM Models. IBM Model 1 is vulnerable in reordering or adding and dropping words. IBM Model 2 has an extra alignment strategy, which says that the alignment of a word in the source (S) depends on where it was in the target (T) and also incorporates foreign words using alignment probability distribution. IBM Model 3 states how many source words a target word can generate by employing the concept called fertility, although the model showed an issue of adding words. In IBM model 4, the alignment of later source words generated by a target word depends on what happened to earlier source words produced by that target word. IBM model 5 is a much more sophisticated model (modified version of IBM Model 4) with added training parameters in order to defeat the non-deficient alignment. In contrast to Model 3 and 4, here, words could be placed only in the vacant positions.

Later, with improvements in SMT, new techniques such as syntax-based [Yamada and Knight, 2001], and phase-based translation [Marcu and Wong, 2002, Koehn et al., 2003] showcased an improvement in translation quality over the IBM model 4 and 5. A more linguistically motivated approach [Yamada and Knight, 2001], (such as one that can handle word order difference) used the parsing tree of the source sentence to transform and generate the target language. The model

[1] In simple words, EM algorithm is an iterative approach that works by repeating two steps: First, it attempts to estimate the missing or latent variables, called the estimation-step (E)and then it attempts to optimize the parameters of the model to best explain the data, called the maximization-step or M-step. These steps are repeated until stability is reached.

parameters were estimated using the EM algorithm. In contrast to previous approaches, a joint probability model for phrase translation by Marcu and Wong [2002] shows the capability of learning phase-based translation as well as the translation at the word level. The work in Koehn et al. [2003] illustrated a similar strategy, but with a limit to the size of phrases up to three word, and with the lexical weighing of phrase translation, which delivered a better accuracy.

2.1.2. Neural Machine Translation

The extensive development in Recurrent Neural Networks (RNN) and machine translation techniques surfaced Neural Machine Translation (NMT) [Kalchbrenner and Blunsom, 2013, Sutskever et al., 2014, Cho et al., 2014, Bahdanau et al., 2014] which demonstrated a competitive performance over SMT. The research work of Kalchbrenner and Blunsom [2013], introduced a single large translation system which takes in the sentence and outputs the translation. The first step is to extract a fixed-length vector representation from variable-length source sentences to train Convolutional Neural Network (CNN) (encoder). Later, the translation step generates variable-length translation with a recurrent language model (decoder). Although their approach does not rely on any alignments, the model was sensitive to the order of the input sentences' words and meaning. This method showcased the best BLEU of 22.5 on all four English to French (EN-FR) News test set of WMT-09 [2] to WMT-12 [3].

Cho et al. [2014] and Bahdanau et al. [2014] proposed a similar architecture of encoder-decoder with gated recursive convolutional neural network (grConv) and novel Recurrent Neural Network (RNN). These two approaches also introduced the concept of attention in NMT. In the research work of Bahdanau et al. [2014], authors applied an attention mechanism in the decoder. This attention module pays extra weightage to the important elements of the input sentence, which relieves the burden on the encoder to carry all the information into a fixed-length vector. They showcased the best BLEU of 28.45 on the WMT-14 EN-FR News test set and 36.15 on the sentences without UNK. Along with the advancement mentioned above, Cho et al. [2014] also addressed the encoder encoding the input to a fixed-length vector. The authors of both the papers mentioned two crucial concerns, one that the NMT model suffers from the curse of sentence length and second, on how to better to handle rare and unknown words (UNK).

Sutskever et al. [2014] proposed an encoder-decoder structure which is similar to Kalchbrenner and Blunsom [2013] but instead of using CNN or grConv, they utilized LSTM [Hochreiter and Schmidhuber, 1997b]. In the architecture, encoder-decoder both used LSTM and delivers a BLEU

[2] https://www.statmt.org/wmt09/translation-task.html
[3] https://www.statmt.org/wmt12/translation-task.html

score of 34.8 on the WMT-14 [4] EN-FR translation task's entire test set. The work showcased a better performance with LSTM on long sentences with limited vocabulary when compared to the previously mentioned approaches, including SMT methods. Although LSTM rectified the dependency issue at some level, it was still a problem for RNNs and CNNs to handle long dependencies and UNK words.

Later, the authors of Wu et al. [2016] proposed a Deep Neural Network (DNN) approach with an 8-layer encoder-decoder LSTM architecture. The research also introduced residual connection and the attention mechanism connecting the last layer of the decoder to the first layer of the encoder. For the UNK word issue, Wu et al. [2016] suggested splitting the words for the input and the translation into sub-word units, which delivered a BLEU score of 38.39 on EN-FR News test set of WMT-14. One of the predominant methods for sequence-to-sequence (seq2seq) learning is the work of Gehring et al. [2017]. This model uses CNNs with Gated Linear Units (GLU) [Dauphin et al., 2017] for both the encoder and decoder, and includes a multi-step attention layer. This model demonstrated the best BLEU score of 35.7 on the EN-FR News test set of WMT-14.

2.1.3. Transformer Models

Until now, we have discussed different NMT models with state-of-the-art RNN/CNN (seq2seq architecture), attention mechanism and sub-word vocabulary strategies. However, in the recent past, machine translation architecture has changed drastically, especially with the introduction of the transformer model [Vaswani et al., 2017]. This new architecture [Vaswani et al., 2017] delivered a BLEU score of 41.0 on the EN-FR News test set of WMT-14.

Vaswani et al. [2017] created the base for all the current state-of-the-art language models. It was the first approach that was not a seq2seq architecture and considered all input tokens (words from a sentence) simultaneously. This architecture entirely relied on stacked-attention and point-wise fully connected layers to build a representation between input and output. They proposed the concept of self-attention where the encoder has access to all the token present in input sentence to encode any token. This method helps the encoder to encode the token with more relevant information and solves the long dependency problem [Bahdanau et al., 2014, Cho et al., 2014, Sutskever et al., 2014] to a certain extent.

Edunov et al. [2018] proposed a transformer architecture but with a different training strategy. The authors used the monolingual corpus to improve fluency [Brown et al., 1990] with a data augmentation technique of back-translations of target language sentences. The use of monolingual data improves the quality of translation, and this work had a BLEU score of 45.6 on EN-FR News

[4] `https://www.statmt.org/wmt14/translation-task.html`

test set of WMT-14.

A novel transformer model, Scaling NMT by [Ott et al., 2018a], showcased an improvement in training efficiency while maintaining state-of-the-art accuracy by lowering the precision of computations, increasing the batch size and enhancing the learning rate regimen. The architecture uses the transformer model with 6 blocks in encoder and decoder networks. This model showcased a BLEU score of 43.2 (with 8.5 hours of training on 128 GPUs) on the EN-FR News test set of WMT-14.

Last year, the paper XLM by Conneau and Lample [2019a] proposed three models, two unsupervised ones that do not use sentence-pairs in translation relation, and a supervised one that does. We have used the third model, named Translation Language Modeling (TLM), which tackles cross-lingual pre-training in a way similar to the BERT model [Devlin et al., 2018b] with notable differences. This approach performed very well on low resource language pairs. For example, for the WMT'16 German-English MT task, it obtained 34.3 BLEU which outperformed last best model with more than 9 BLEU and for WMT'16 Romanian-English it obtained a BLEU score of 38.5, which outperformed second best model with more than 4 BLEU points.

In the two papers from Chapter 3 and 4 we have chosen NMT models [Gehring et al., 2017, Ott et al., 2018a, Conneau and Lample, 2019a] to evaluate the performance of our approach and for translation as well.

2.2. Cleaning Translation Memories

This section discusses the state-of-the-art research for cleaning translation memories. We start our discussion with statistical approaches like different scoring methods and intricate feature designing and how it laid the foundation for the current state-of-the-art methodologies.

There are many approaches which showed a great deal of success in cleaning TM as discussed in-depth in this Section 2.2.1. First we discuss the scoring-based techniques such as Brown et al. [1991], Gale and Church [1993b], Imamura and Sumita [2002] and Imamura et al. [2003] from which we have taken some ideas for our work described in Chapter 3. The research later focused on using IBM models which was tried with bootstrapping methodologies elaborately discussed in 2.2.2 such as, Fung and Cheung [2004a], Fung and Cheung [2004b], Fung and Yee [1998] and Ananthakrishnan et al. [2010].

With the development in machine learning models, especially feature-based supervised classification models, research shifted its focus towards designing linguistically driven features. More advanced practices such as, building word representation in multidimensional vector space for semantically driven features became a common technique. In Section 2.2.3, we will discuss aforementioned feature driven approaches, such as, Barbu [2015], Buck and Koehn [2016], Jalili Sabet et al. [2016a], Nahata et al. [2016] and Chu et al. [2014]. Similar to machine translation, neural networks also have been used for TM cleaning task [Chu et al., 2016, Espana-Bonet et al., 2017] and complex deep networks were also utilized [Grover and Mitra, 2017, Luong et al., 2015, Grégoire and Langlais, 2018] which are described in Section 2.2.4. Similar to the WMT translation tasks, there are shared tasks on corpus cleaning in the WMT (for the past few years), some of the popular approaches being Wang et al. [2018], Rossenbach et al. [2018], Khayrallah et al. [2018], Bernier-Colborne and Lo [2019] and Chaudhary et al. [2019] as discussed in Section 2.2.5.

2.2.1. Statistical methods based on generating scores and features

In this section, we review the research based on statistical methods associated with the size of the sentence (word & character level), including one of the popular word alignment and translation tools IBM models.

One way to clean a TM is to identify Sentence Pairs (SPs) that are poorly aligned. Initial research [Brown et al., 1991, Gale and Church, 1993b] aligned sentences based on the length of the SPs at the token level and character level, respectively. In Brown et al. [1991], they worked on aligning the Canadian Parliament Hansard corpora by identifying correspondences between SPs, based on the number of tokens, but making no use of lexical details.

Gale-Church score [Gale and Church, 1993b] is a probabilistic score based on the difference of length and the variance of the difference between the source and proposed correspondence sentence in the target language. The authors employed this score to determine the maximum likelihood alignment of sentences. This method aligned 96% of the Union Bank of Switzerland's data on different languages such as English (EN), French (FR), and German (DE). Although the experiments were successful, translation quality was not considered (along with different noise). This research showcased how the length of the source and translation sentences is an essential contributor in determining the quality of the translation, but thanks to Barbu [2015] we later understood it is merely one of the many vital aspects.

Scores like Gale-Church score alone can not solve this problem; therefore, IBM models were proposed to generate automatic features from an SP. An exciting research work by Munteanu et al. [2004] has been proposed, where the authors extract sentences (monolingual) from two monolingual comparable newspaper corpus (Gigaword) in Arabic and English. In Munteanu et al. [2004], the authors trained a Maximum Entropy (ME) classifier to classify whether a pair of sentences is parallel or not. The authors applied IBM Model 1 for word alignment. They designed features from the length of the sentences, the total number of words in both source and target, word percentage with no connection, alignment score, and few others. The authors then employed the extracted SPs (parallel) to demonstrate the improvement in the translation quality of a baseline SMT system.

Munteanu et al. [2004] agrees that the extracted SPs could contain noise such as sentences with small differences in content, but the SMT model is not prone to such noises. Also, the extracted data never outperformed the SMT system trained on high-quality data. In Chapter 3, we have explored a similar approach where the output of our classifiers, a “cleaner corpus" was fed to the NMT systems to measure the improvement in translation quality.

A popular research [Munteanu and Marcu, 2005a] proposed to extract parallel data from Chinese, Arabic, and English non-parallel newspaper corpora. In contrary to previous work [Munteanu et al., 2004], better features and a much more advanced SMT system were used to evaluate the ME classifier performance. With a robust system in hand, there are still many issues regarding the spurious translations and uncommon words. The IBM Model 1 does not consider word order, i.e. it can connect one source to arbitrarily many target words, hence creating incorrect links as features for the classifiers. According to the authors, this approach is not suitable for the data where SPs share many useful links but express different meanings (False friend).

It is hard to detect SPs which are semantically close but showcase different meaning (occur due to wrong lexical choice during translation) we tried to address such problems in Chapter 3. The meaning (at some level) was captured by our robust deep learning classifier (bi-LSTM), which consumed parallel SPs also a closely related non-parallel SPs as bad examples. Such negative sample generation is explained in detail in Chapter 3.

The extension of the work Munteanu and Marcu [2005a] came a few years later with Tillmann [2009], where the authors used the beam-search algorithm to filter out target sentences, and the maximum-likelihood classifier applied on the filtered data. The new concept of introducing a beam-search filtering improved a BLEU score by one point on their Spanish-English data.

The work in Nie and Cai [2001] demonstrates to clean a web scraped data by improving the translation accuracy of the resulting translation model and the effectiveness of Cross-Language Information Retrieval (CLIR) using these models are also improved considerably. The authors based their approach on the following three factors: length, empty alignments and known translations. Empty alignment means, there was no translation for the source web documents and a bilingual dictionary was used for evaluating known translations. Based on the known translation they evaluate a degree of correspondence which is then integrated with the Gale-Church score. The evaluation of translation quality was done through a set of 200 randomly selected words in

Chinese and English, and the authors examined the first translation of these words by the models. This step showcased an improvement of respectively 12.27% and 14.94% for English-Chinese and Chinese-English.

The research works such as Imamura and Sumita [2002] and Imamura et al. [2003], focused on clean bilingual corpus (English-Japanese) intending to improve the SMT translation quality. Apart from removing noisy SPs, the research work Imamura and Sumita [2002] also removes SPs where one source sentence has more than one type of target translation from TM to build a better translation knowledge. Very different from previous approaches like Gale and Church [1993b] and Munteanu and Marcu [2005a], the authors of Imamura and Sumita [2002] proposed word-level and phrase-level literality scores which decides whether to drop an SP or not. The authors dropped almost 19% of the SPs from the TM, and it showcased the improvement in the machine translation quality to about 87% (subjective evaluation).

Another work Imamura et al. [2003], proposed to clean TM by removing the SPs based on the concept of the hill-climbing algorithm [5] (search for the combinatorial optimization). The authors trained the SMT system with all SPs and built several rules, and eventually, incorrect/redundant rules were removed based on the feedback BLEU score from the SMT. This method also showcased an improvement by 10% in machine translation quality (subjective evaluation) and a BLEU score gain of 0.045.

The authors of Khadivi and Ney [2005] approached the problem by using the improved version of the Gale-Church score, with additional features like penalty score based on dictionary and match phrase in source exists in the target sentence. With this method, the authors keep 97.5% of the corpus and showcased an improvement in BLEU score of 0.4%.

2.2.2. Bootstrapping methods to extract parallel texts

Similar to corpus alignment, extracting only parallel SPs from any corpus also solves the purpose of "Cleaning Translation Memories" by marking the rest as "noise". The research [Fung and Cheung, 2004a,b] build lexical features using IBM Models 4 and also by an Information Retrieval (IR) technique to generate bilingual lexicons [Fung and Yee, 1998].

The research work of Fung and Cheung [2004a], first extracts the similar documents by applying similarity-based features. The authors later exploit the bootstrapping method to extract parallel sentences from a very-non-parallel corpus (English-Chinese). The main idea is that if one parallel sentence pair is found in the document, it must have more such pairs. Bootstrapping was used

[5] `https://en.wikipedia.org/wiki/Hill_climbing`

to extract more parallel sentences from such documents. This method also tries to find parallel sentences, even if the documents are not similar enough.

The authors employed IBM Model 4 EM lexical learner to update the dictionary for unknown translations using the extracted parallel SPs. The method produced 67.5% accuracy, with 50% relative improvement from the sentence similarity score (baseline). This paper also concluded that the lexical learner does not perform well on a very noisy corpus and to boost this performance, the authors consider a bootstrapping method.

Later that year, an unsupervised learning approach [Fung and Cheung, 2004b] proposes to extract parallel SPs from Quasi-Comparable corpus of different sizes, which also contain in-domain and out-of-domain documents. Similar bootstrapping methods as Fung and Cheung [2004a] were applied with a different hypothesis, which claims that an efficient comparable-documents extractor leads to better SPs and vice-versa. Bootstrapping this iteratively and efficiently leads to comparable-documents, sentence pairs, and bilingual lexicons. Rather than using IBM Model 4 EM, a bilingual lexicon was extracted from parallel sentences using the method proposed by Fung and Yee [1998] also named-entity method was used to build lexicons. Although, in the end, the authors reported a similar accuracy as the one mentioned in Fung and Cheung [2004a].

Ananthakrishnan et al. [2010] proposed a method with a bootstrapping approach, where the goal is to re-sample the corpus by evaluating the quality of the phrase pairs. This approach delivered an improvement in translation quality with a 1.7 BLEU score jump on an SMT system trained on English-to-Pashto language pair.

2.2.3. Advancement in Feature engineering

A few years later, an entirely different strategy by Barbu [2015], Jalili Sabet et al. [2016a], where the authors used several engineered features to capture the semantic relationship between the SPs. These approaches proved a valuable advancement over the innovations discussed earlier and demonstrated better results. Both the authors worked with the largest TM MyMemory [6] which contains more than 1 billion SPs for 6000 language pairs and tends to have noise and irregularities.

Barbu [2015] tried to spot false translations by building a classifier using the Gale-Church score. The model also used 17 derived features such as URL matching, tag flag, number match flag, punctuation similarity, presence on capital words on both sides. These features were built based on the authors' understanding of the issues in MyMemory. The classifiers such as SVM and Logistic Regression performed best among others with F1 scores of 81% and 80%, respectively.

[6] `https://www.mymemory.com/`

They also conclude that the classifier ended up with high false-negative value, which means this method deletes valid parallel sentence pairs from MyMemory and it is not sufficient for automatic cleaning of translations memories.

The approach proposed by Chu et al. [2014] is in a similar direction as Barbu [2015], where they address the parallel corpus extraction from Wikipedia for Chinese-Japanese language pair. The authors used the features introduced in Munteanu and Marcu [2005a] along with the Chinese character-based features, and in the end word alignment by GIZA++ [7]. SVM classification model was employed, similar to the work described in Munteanu and Marcu [2005a]. The classifier uses false translation as the Cartesian product of the actual translations; such negative samples only consider misaligned noise. In Chapter 3, we do showcase the SVM classifier but using systematically generated negative samples to handle subtle noise.

We will now discuss the three unsupervised learning approaches to clean the TM as proposed by Taghipour and Khadivi [2011], Cui et al. [2013], Jalili Sabet et al. [2016a]. Taghipour and Khadivi [2011] approached this problem with outlier detection, an unsupervised learning approach that uses kernel-based techniques and K-nearest neighbours. The authors designed features from translation model probabilities using IBM Model 1, N-gram language, sentence length, word-alignment features from Munteanu and Marcu [2005a]. These features were then employed to estimate density, and with lower density, model annotate SPs as wrong translation. They utilized SMT phrase-based model proposed by Koehn et al. [2003] to evaluate the improvement in the translation quality with a gain of 0.6 BLEU score.

Cui et al. [2013] introduced an unsupervised approach using a graph-based random walk algorithm. The authors derived the vertices from sentences and phrases, and the edges denote the importance. The model computes the importance measure recursively and filters out low-scored (low-quality) SPs; that is, the SPs with non-literal translations are based on low frequency. The performance was measured based on the improvement in a BLEU score of 0.47 on their English-Chinese trained SMT model.

Jalili Sabet et al. [2016a] (a.k.a TMOP) proposed an unsupervised method to clean the English-Italian (EN-IT) translation unit (TU) randomly picked from MyMemory. This approach automatically creates training labels for a subset (11M TU) using the similarity-based features evaluated between the source and the target. The authors extracted the necessary features from the work of Barbu [2015], and Quality Estimation QE-derived features from the word alignments done by MGIZA++. They also utilized the cross-lingual word embeddings of 100 dimensions vector for each word, which generates features like cosine similarity and average embedding

[7] `http://code.google.com/p/giza-pp`

alignment score. Each group of these features has a specific task, either to detect bad alignment, bad translation quality or "semantic" distance between the SPs. They outperformed the approach given by Barbu [2015], where Barbu [2015] trained a classifier on human-labelled data. Although the results were comparatively better, they evaluated the model on a small sample of 1000 TU, which gave approximately 79% accuracy with no information about false positives.

In Chapter 3, we employ TMOP [Jalili Sabet et al., 2016a] by utilizing the GitHub [8] implementation. Based on our results in Chapter 3, cleaning TM using just these features could produce misclassification of closely related good and bad SPs.

The first “Automatic TM Cleaning Shared Task” [Barbu et al., 2016] competition published three popular approaches Zwahlen et al. [2016], Buck and Koehn [2016] and Nahata et al. [2016]. The shared task evaluates three classes, i.e. true translation, true translation with minimal error (minor post-editing is required), and wrong translation. The work such as Zwahlen et al. [2016] ,Buck and Koehn [2016] and Nahata et al. [2016] are feature-based approach to detects false translation and also suggests if certain SPs require some minor editing. A rule-based classifier [Nahata et al., 2016] which proposed to detect accurate translation in the TM for languages EN-DE, EN-IT, and English-Spanish (EN-ES). The authors designed these rules from sentence length, first character of source and target, capital letters, numbers matching, length of the longest word in source and target, presence of different punctuation, and end delimiter matching.

The approach of Zwahlen et al. [2016] is based on the 17 features from proposed in Barbu [2015] with additional features generated using POS tagging. The classification model framework was also based on the work of Barbu [2015]. They used features like number of tokens, characters, alphanumeric and digits and other complex features like probability score from 5-gram language model (source and target), word-alignment model score from `fast_align` [Dyer et al., 2013], and probability score from NMT model using subword units [Sennrich et al., 2016]. Similar to Zwahlen et al. [2016], work in Buck and Koehn [2016] also mentioned Random Forests model performed best for the three class classification task shared by Barbu et al. [2016].

The work of Barbu [2017] proposed to ensemble multiple classifiers and compare the performance of the individual models and the ensembles model, which was not the case with Barbu [2015], Buck and Koehn [2016] and Zwahlen et al. [2016]. The authors applied Majority Voting, Stacking, AdaBoost and Bagging with the features from the Barbu [2017] and stated that “this method is suitable for the data crawled and aligned from the web parallel sites" .

[8] https://github.com/hlt-mt/TMOP

In Chapter 3, we have designed some of our features based on the work of Barbu [2015] and Nahata et al. [2016] for the first step cleaning TM. We conjoined the noise and supposedly clean samples detected in this step to generate artificial noise systematically, which was used to train deep learning and feature-based classifiers for the next step of cleaning.

2.2.4. Deep Learning based approach

With the learning from previous methods, the authors of Chu et al. [2016] applied Neural Network (NN) based features. The method demonstrated a minor improvement but seeded the idea for further experiments. Their work was an enhancement over the previous Chu et al. [2014] work which was a pure statistical approach. It retained all the steps, including the generation of negative samples for binary classifiers. The additional step was to train an NMT model using the extracted SPs. The trained NMT model was then employed to generate additional features such as BLEU scores, and combine it with the features from Munteanu and Marcu [2005a] to train an SVM classifier.

In order to understand the meaning of SP in a cross-lingual setting, a robust semantic relationship needs to be built, which lacked in the previous studies. In Chu et al. [2016], although NMT models were used to generate the sentences, it lacked the cross-lingual comparison. Another big step proposed in the work of Espana-Bonet et al. [2017] was building an advanced NMT model. Espana-Bonet et al. [2017] trained NMT models for the six different language pairs using 56M parallel sentences with 60K fixed vocabulary (many more versions with different vocab and language pair). The motivation behind their work was to represent similar sentences (within and across languages) together in a vector space. The authors derived features from character N-gram and pseudo conjugate and also included basic features like length at word and character level. The SVM classifier was trained on these features and they reported an F1 score of 98.2% in the shared task at "Building and Using Comparable Corpora" (BUCC) 2017.

We employed a similar approach mentioned in Espana-Bonet et al. [2017] in Chapter 3, where we used a pre-trained language model, LASER [Artetxe and Schwenk, 2019a] to get the cross-lingual sentence representation. A different scoring technique was utilized to measure the alignment between the representations of source and target sentences. This setting was also kept entirely unsupervised in the study to compare the performance with our robust supervised learning approach. The work in Artetxe and Schwenk [2019a] was also the best performing method in the BUCC-2019 mining task.

In the previous research, the focus was not on learning sentence representation in high dimensional space. Rather than calculating the similarity scores from the semantic features (previous methods), Neural Networks (NN) can generate and learn a sophisticated feature space. The initial work in this field was proposed with Grover and Mitra [2017], where the model learns the bilingual word representation proposed by Luong et al. [2015] to form the similarity matrix between the words for each SPs. It was the first time a CNN learnt the sentence embedding obtained from a NN word embedding technique. The embedding technique in Luong et al. [2015] was based on the novel skip-gram method [Mikolov et al., 2013b] but utilized both monolingual (to capture text concurrence information) and bilingual data (for meaning equivalent signals).

An innovative work was proposed by Grégoire and Langlais [2018], where the model tried to estimate the conditional probability distribution that the SP is a true translation. They demonstrated how a single end-to-end model can exploit artificially created negative SPs (randomly sampling 10 false translations for every correct translation) to develop a parallel sentence extraction system. The classification setting first generated sentence representation (from Word2Vec [Mikolov et al., 2013a]), which was then fed to bi-LSTM encoders (source and target). Later, combined the two encoders' output by summing the dot product and difference between the two. The model trained on the EuroParl corpus and artificially created negative SPs by randomly sampling 10 false translation for every true translation. This work showcased an improvement in translation quality (BLEU Score) measured by both SMT and NMT system (OpenNMT [9]).

We used this approach in Chapter 3 as a cleaning technique with different parameter settings, also, rather than adding clean SPs to show improvement in BLEU, we removed the noisy SPs. The future work suggested in the original paper [Grégoire and Langlais, 2018] to replace the sentence (either source or target) with its nearest neighbours (for negative sampling) was examined and it improved the classifier's performance. In Chapter 3, we had used the in house trained state-of-the-art NMT models.

2.2.5. WMT 18 & 19 Corpus Filtering Task

In 2018-19 WMT [10] a popular machine translation conference introduced the "Parallel Corpus Filtering task" which inspired this thesis in many ways. In WMT-18, the dataset assigned was German-English corpus crawled from the web (part of the Paracrawl project) with 1 billion words. The work of Wang et al. [2018] proposed to de-noise corpus by online data selection which showed an improvement in BLEU by $+7.5$ on WMT-2018. The model intends to remove noise and adapt the domain, but it can hurt out of the domain test sets. The approaches we proposed in Chapter 3 are not domain-specific, and our supervised learning module showed a gain of +1.27 BLEU for the corpus with 200 different domains.

[9] `https://github.com/OpenNMT/OpenNMT-py`

[10] `http://www.statmt.org/wmt18/,http://www.statmt.org/wmt19/`

The authors of Rossenbach et al. [2018] followed a similar set of steps we did in Chapter 3, where noise based on the rules (designed by a professional linguist) was filtered out, and then more sophisticated filtering techniques were implemented on top of the remaining SPs. The authors first filtered SPs with tokens seen less than three times and applied heuristics based on sentence length, Levenshtein distance, token work ratio, and redundancy. Later, the authors used the joint scoring (log probabilities) from different language models such as IBM Model 1, KenLM count-based model and transformers [Vaswani et al., 2017] to score each SP. In Chapter 3, the designed heuristics were more sophisticated compared to ones presented in this approach, and we trained our classification models (filtering methods) with the intention to detect noisy SPs of wide range.

Khayrallah et al. [2018] proposed a modification of the toolkit demonstrated in Xu and Koehn [2017], that assigns quality estimation score (bag of words translation score) and fluency score (based on 5-gram KenLM language model) to SPs. Based on these two scores, the authors train a logistic regression classifier. They delivered the best BLEU score of 30.20 in the WMT-18 task (NMT 100 million words).

We have not utilized this scoring method in Chapter 3, as the method highly depends on customized dictionary and n-gram models from an aligned EN-FR corpus. The noise detected by our proposed method in Chapter 3 is way more complex to be comprehended by such scores, although we have not conducted any such experiment to prove that.

A TM cleaning tool Bicleaner [11] (similar to Xu and Koehn [2017]) which is based on hard rules such as flag foreign language (language different than English-German), encoding errors and the difference in length of SP. The previous approach, such as Khayrallah et al. [2018] did clean the corpus, and we should also note that the test corpus (newstest18) was very noisy. As mentioned in Khayrallah et al. [2018], the BLEU on randomly selected SPs was close to 9.2 and after cleaning achieved the best BLEU of 34.8. We had measured a BLEU score on uncleaned data randomly selected as 36.25 (from Chapter 3), which is way higher; therefore, it is also tough to clean an already cleaned TM.

Another feature-based approach was that of Lu et al. [2018], in WMT-18, where the authors first filtered the TM based on rules such as length ratio of source and target sentences, edit distance between source and target tokens, URL-match. The filtering was then carried by different bilingual scoring based on word-alignment and Bitoken CNN Classifier [Chen et al., 2016], language model scores, and N-gram-based diversity score. The authors reported the best BLEU score of 31.44 for the WMT-18 task.

[11] https://github.com/bitextor/bicleaner

Another approach by NRC [Lo et al., 2018] in WMT-18 shared task, proposed a semantic-based SP scoring technique `Yisi` [Lo, 2019] which takes into account both the monolingual and cross-lingual semantic and trained a linear model with L1 regularization to train the classification model. This simple approach demonstrated a BLEU score of 31.76 on average for the tasks.

The NRC submission in WMT-19 [Bernier-Colborne and Lo, 2019] consists of four different approaches, the first two were similar to WMT-18 [Lo et al., 2018] with `Yisi` [Lo, 2019] but the authors purely relied on cross-lingual lexical semantic similarity from the bilingual word embedding. The third method used a deep transformer approach XLM [Conneau and Lample, 2019a], where the transformer was pre-trained on the low resource WMT-19 data (Nepali, English, Sinhala and Hindi) and fine-tuned on a classification task. The authors also induced artificial noise by generating four negative samples (from the cartesian product of source and target) for each parallel SP as a training sample. An ensemble of `Yisi` and XLM classification performed well for this low resource task. The inspiration to use XLM as our evaluation and machine translation engine in Chapter 3 and 4 respectively originated from this paper.

A very robust RNN model named LASER [Artetxe and Schwenk, 2019a] is a single encoder designed and trained to handle semantically similar sentences in different languages close in the embedding space. It assigns a score using the method “margin", a ratio that relies on the SP's nearest neighbours. This architecture was used in WMT-19 [Chaudhary et al., 2019] where authors ensemble LASER with other methods like Zipporah [Xu and Koehn, 2017], Bicleaner [Sánchez-Cartagena et al., 2018] and Junczys-Dowmunt [2018] to score each SP. The WMT-19 "Parallel Corpus Filtering task" was based on the low resource language pairs (Nepali-English, Sinhala–English), and the authors trained a LASER encoder on it. Due to lack of data, authors exploited other cleaning methods to score each SP. In the end, the authors did mention that LASER-toolkit [12] was tested, which is trained on 93 different languages was tested and demonstrated a BLEU score gain of 0.4 on the ensemble method mentioned above. The authors mentioned that Nepali was not even part of the 93 different languages used in training LASER.

Chaudhary et al. [2019] showcases that we can deploy the LASER-toolkit without training or fine-tuning, therefore in Chapter 3, we had focused in-detail on this approach and employed it as an unsupervised learning strategy. We showcased a competitive BLEU score gain when compared to the supervised learning method.

[12] `https://github.com/facebookresearch/LASER`.

2.3. Human and Machine Translation Detection

Now that we have a decent idea about needs and methods for cleaning TM's, the focus of this thesis shifts towards understanding human- and machine-translated text. In the below section, we first discuss the evolution of translation detection tasks where professional translators did the translated text [Toury, 1980, Baker et al., 1993, Blum-Kulka and Levenston, 1983, Laviosa-Braithwaite, 1998, Volansky et al., 2015]. Later with the advancement of machine translation systems, the research shifts towards Machine Translation Detection (MTD) [Carter and Inkpen, 2012, Antonova and Misyurev, 2011, Arase and Zhou, 2013, Aharoni et al., 2014, Nguyen-Son et al., 2019b, Juuti et al., 2018]. The MTD task first started with the approach discussed in Section 2.3.1 and later with more complicated features- and models' architectures.

2.3.1. Human Translation Detection

In order to detect (human) translation, one needs to understand how to differentiate the native (original) and translated text. Initially, the studies were done in the direction to understand and extract features from the human-translated text, although such studies with the later development became more and more uncertain.

One of the initial studies [Toury, 1980] talks about the concept called "Universal Translation Behaviour or Translation Universals" (translationese), which discusses generic rules and norms that can separate translated text from the native texts. Similarly, Baker et al. [1993] and Blum-Kulka and Levenston [1983] suggested the hypothesis that there exists a set of "universal features" between translated and native text. Toury [1980] also suggest these features exist because of the constraints in the translation process. The work of Laviosa-Braithwaite [1998] states in the Encyclopedia of Translation Studies (1998) that the "Translation Universals" occur in the translated text rather than native text, and it is independent from the languages involved.

The author of Laviosa-Braithwaite [1998] selected a fixed number of features for all types of translation based on contrastive analyses of the source and the translation. Some of these features are avoidance of repetition in the source text by Toury [1980], simplification such as lexical, syntactic and stylistic ones were identified in the translated text by Blum-Kulka and Levenston [1983] and disclosure transfer a.k.a "Law of Interference" given by Toury [1980]. The "Law of Interference" captures a positive or negative interference in the translated text when a translator tailors the source text structure to the target text. The negative interference means the translated text varies from the general norms defined by the target language rules.

Over the time, there were a few contradicting theories proposed in Toury [2004], Chesterman [2004], Pym [2008] and Malmkjaer [2008]. According to Pym [2008], if "Translation Universals" exist not only for translated text but also for other texts, then it is not a “universal”. Also, the following studies Puurtinen [2004], Saldanha [2008] and Becher [2011] contradict the presence of “universal features” for every translation, and the resultant development was not applicable for all types of text.

According to Toury [2004], processes/models just based on hand-picked features were hard to interpret. Therefore, the work [Baroni and Bernardini, 2006] suggests that if the difference between the translated and native texts is big enough for a large data set (training set), a machine can learn to identify such a pattern. The paper [Baroni and Bernardini, 2006] supports the translationese hypothesis (but not as “universal”) by designing a classification model with SVM. The features were derived from the distribution of words that build a grammatical relationship and other grammatical words like nouns, pronouns, adjectives, finite verbs, auxiliary verbs, adverbs, Parts of Speech (POS) ngrams. The authors conducted the experiments on a monolingual corpus to identify translated text from Italian articles (in the geopolitical domain) and showcased a best accuracy of 86.7%.

On similar lines, Kurokawa et al. [2009] considered an EN-FR TM (Canadian Parliament Hansard corpora) with reference information and delivered 90% accuracy in detecting translated text from native. The authors also used similar features and the classification model discussed in Baroni and Bernardini [2006]. This work suggests that it is possible to distinguish native text and translated text from a different language. The authors state that the SMT performance increases if we only train the model with the data that does not have a mix of native and translation data.

The research work [Ilisei et al., 2010] has been proposed based on the characteristic features built from the length of the sentence and words, parse tree, presence of simple, compound and sentences with no finite verbs. The authors derived the rest of the features from the work of Baroni and Bernardini [2006]. The paper showcased an accuracy of 87.16% using an ensemble model of Decision Tree, Logistic regression and Jrip on monolingual Spanish corpora from the medical domain.

Koppel and Ordan [2011] showcased that, rather than identifying translated text from a fixed source, it is possible to identify translated texts from multiple languages by training a classifier on the data with a fixed source. The approach [Koppel and Ordan, 2011] delivers an accuracy of 97.6%, which is in line with previous approaches [Ilisei et al., 2010, Kurokawa et al., 2009], but here, the authors showcased this result on test data from different source languages. Apart from general translationese features discussed above, the authors used features based on the frequency of animate pronouns, possessive pronouns, presence of cohesive markers.

In Volansky et al. [2015] authors uses features like lexical variety (assumptions with native text is rich in vocabulary), the average length of word and sentence at character level (assumptions with translated text contains simpler words) and other features like syllable ratio (less syllable words in the translated text), lexical density, cohesive markers and frequency-based features. The authors also used features derived from the work of Baroni and Bernardini [2006].

At the end of this section, we conclude that linguistically informed features can identify translated text written by professional translators (human), and the method can generalize to different languages. Although, with these features in hand, one can design a classification task with sophisticated machine learning algorithms, none of the research showcased a perfect accuracy for the task; hence we can assume that these are not “universal features” but good indicators.

2.3.2. Machine Translation Detection (MTD)

In this section, we discuss the machine translation detection task, starting with the low-quality machine translated data mostly produced by SMT systems, then wrapping up with the high-quality neural-based machine translated data detection methodologies.

2.3.2.1. *MTD on low-quality machine (SMT) translated extracted data from web*

Carter and Inkpen [2012] proposed an approach to detect poor-quality machine-translated text (generated from Microsoft’s Bing Translator of that time) from the human-written text, i.e., native and translated by a human. An SVM classifier was trained on simple features like unigram frequencies and length of the sentence and achieved an accuracy of 99.8% on the Canadian Parliament Hansard corpora. The authors also state that the model accuracy falls for the out-of-domain test set and generates a high rate of false positives. In the end, the authors also mentioned that, one translation detection model would not be enough to detect translation data generated from different machines.

During this time, most of the researchers focused on cleaning web-extracted TM to train better SMT models, and the obvious noise detected was the machine-translated text. For example, Rarrick et al. [2011] demonstrates the improvement of machine translation quality by detecting and removing machine-translated text from web scraped TM. The authors designed features from the number of tokens and characters, URL match, number of out-of-vocabulary tokens on the source and target sentences, and at document level number of aligned sentences and alignment scores. The authors did notice some improvement in BLEU score of 0.59 for English-Latvian. The noted gain was for limited test sets; most of the language pairs did not demonstrate any gain. Hence, the authors suggest to extract much cleaner SPs by adjusting the threshold of the classifier.

In Antonova and Misyurev [2011], the authors first extract the parallel corpus from the web and remove machine-translated text. They proposed a similarity algorithm based on a unique phrase-based decoder. The authors calculate r-BLEU, which is calculated based on n-gram similarity for the SPs with the reordered references. The authors mark an SP as machine-generated based on a threshold of r-BLEU with high precision (94.1) and recall (90.1). This method showcased an improvement in phrase-based SMT quality measured by a gain of 0.5 BLEU score.

Arase and Zhou [2013] proposed the detection of low-quality monolingual Web-text translated by SMT model [Bansal et al., 2011], which focuses on an ill-formed sequence of phrases that typical SMT system often produce. Their approach used language models trained with POS sequences and 4-grams. This model helps to determine the non-contiguous phrases (i.e. based on inter-phrases sentence connection), which are most common in human translation but not so much in SMT. This method delivered an accuracy of 95.8% and 80.6% for Web-text with noise using SVM classifier.

2.3.2.2. *MTD on publicly available corpora*

Aharoni et al. [2014] designed features that capture the presence or absence of part-of-speech tags and function words taken from LIWC [Pennebaker et al., 2001] appearing at least ten times in the training material. They reports that the accuracy of detecting human or machine translations is inversely correlated to the quality of the translation engine used. The data collection involved a corpus extracted from the Canadian Hansards, and various translation engines (such as Google Translate, Systran [13], commercial machine translation engine [14]) to generate machine translated data. The best model reported an accuracy slightly over 60% with an SVM classifier.

Li et al. [2015b] used a parsing tree to generate linguistically-motivated features using Europarl data for positive samples (human translated data) and translated the sentences using a phrase-based SMT model for negative sample (machine-generated data). As we have seen so far, most of the features focus on the translation side (human and machine translated). They state that the parsing structure of the translation side is very sensible to the quality of SMT outputs. Other features derived were the number of right- and left-branching nodes for constituent types and also for the Noun Phrases, numbers of pre- and post-modifiers, number of adjectives before and after nouns, ratio of count tokens in pre- and post-modifiers for all constituent types and Noun Phrases. The authors trained an SVM classifier, showcased an accuracy of 74.2%, and noted an improvement 1.6 BLEU score on the phrase-based SMT model after removing machine-translated SPs. The authors here did not try to remove all the machine-translated sentences, but just the poorly translated ones from the SMT model.

[13] https://translate.systran.net/translationTools

[14] http://itranslate4.eu

With the advancement in this space, Nguyen-Son et al. [2017] proposed a complex and linguistically motivated feature engineering technique for the MTD task. The authors talk about Zipf's law where, the most frequent token from human translated text is twice the second most frequent tokens and three times the third, but its machine-translated texts does not follow this law. For designing the features based on the frequency of tokens, the authors plotted the regression line and evaluated the information loss. Apart from this, phrase-based features were designed from human translated text using parsing trees, such as idioms-, ancient- and dialect-phrases and some coreference resolution features. The data explored was from Project Gutenberg [15] a source of free online books. The authors translated (into English) around 100 books using Google Translate for machine-generated text. This approach demonstrates an accuracy of 98% using SVM classifiers.

The coherence based approach proposed in [Nguyen-Son et al., 2018] classifies paragraphs rather than each sentence. According to the authors, a machine-translated text looks like human-translated (hard to differentiate) but using different words. The coherence features were created from similar words matching within the sentences (separated each sentence in the paragraphs) and added a penalty score to reduce the effects of unmatched words. In this work, the coherence score is low for machine-translated paragraphs. The authors evaluated the model on 2000 human- and 2000 machine-translated paragraphs, which showcased 72.3% accuracy.

The authors also proposed a similar approach in [Nguyen-Son et al., 2019b] by using the same architecture but with a modified scoring technique. This work performed the word matching at the paragraph level, rather than at sentence level proposed by Nguyen-Son et al. [2018]. For scoring, the authors used euclidean distance evaluated from the word embedding generated using GloVe [Pennington et al., 2014]. This work showcased an improved performance with an accuracy of 87.0% from the previous best 72.3%.

We have also noticed similar behaviours in machine-translated sentences in Chapter 4. According to a native speaker (French) and language expert (Professor Philippe Langlais), it was tough to differentiate the two (human versus machine). The analysis section in Chapter 4 discuses on different possible factors on which our detection models base their decision.

Until now, the focus was on building complex linguistically-motivated features, but Nguyen-Son et al. [2019c] use a simple similarity measure between a sentence, and its back-translation. They distinguishes original sentences (EuroParl) from translations produced by Google Translate and delivers an accuracy of 75% with an SVM classifier. According to the author, the back translations of translated texts should be less modified (low variance) than back translations of original (not translated) ones. The authors measured the similarity with seven variants of a BLEU score,

[15] `https://www.gutenberg.org/wiki/Main_Page`

including individual BLUE for N-gram range 1-4 and cumulative N-gram with range 2-4.

To the best of our knowledge, the work of Nguyen-Son et al. [2019c] is the only approach that uses a neural approach (Google Translate). Previous works focus mostly on building complex features and used classification models like SVM and Decision Tree on SMT output. The space for research in the direction of using deep-learning models was an opportunity and a challenge. We tried to fill the gap by deploying state-of-the-art NMT and classification models (different transformer architectures).

2.3.2.3. *MTD for Plagiarism detection*

Plagiarism is considered a severe offence in every section of our society. In the context of machine translation, as these models got better, it was made public, and with all its boon, there were few downsides too. One such downside is the use of automatic translation by language students at Language teaching institutes for their home works (when prohibited by the language institutes). We agrees that the use of automatic translation by language students is very helpful for them. It acts as a platform to learn new languages, but the students should not misuse it to complete their homework.

With the easy availability of the automatic translation system, online, the language institutes were facing a hard time judging students performance, especially the weak ones. Somers et al. [2006] and Steding [2009] focus on plagiarism, basically detecting machine-translated text in the assignments of the language training class. The authors based their research on finding mistakes in the machine-generated text, which is not frequent by a language student. Somers et al. [2006] gathered the first set of data with the translation done by a group of students where they can take help from books and dictionaries. Furthermore, the second group of students translated using an online translation engine with minimal changes (tidy it up) to look legitimate. Some of the features used were token counts, hapax legomena with the hypothesis that a significant overlap of uncommon words suggests copying, n-grams (up to 9) and based on BLEU and Levenshtein distance decision were made. Based on this, they flag (might be plagiarized) the documents, which should be looked at more closely.

One off-topic work, but worth noticing, Juuti et al. [2018], proposed an approach to detect fake reviews by systematically generating context-based fake (on-topic) reviews from an NMT model. For any restaurant, online reviews play an essential role in attracting new customers, but if these reviews are generated automatically (fake), it is a threat to the credibility of this online platform. The authors considered the Yelp dataset [16], and generated the fake reviews from a character-based LSTM model (with induced grammatical errors). This paper used the AdaBoost classifier to

[16] `https://www.yelp.com/dataset`

identify real and fake reviews. The authors did mention that the fake reviews generated by their model was hard to be recognized by a native English speaker, which we have also noticed during our experiments on neural translation in Chapter 4. Also, the research work [Nguyen-Son et al., 2019a] in a similar direction detect adversarial text generated by the machine which uses the architecture of Nguyen-Son et al. [2019b]. The work mentioned above helps the organization and our community to be vigilant against the cyber-crime.

All the approaches we reviewed so far for MTD tasks were different in their ways, but almost all the research uses the low-quality machine translation systems (mostly SMT systems), and even the test sets were small (small size and less diverse). In the paper, “Human or Neural Translation?” discussed in Chapter 4 we try to address this gap by utilizing state-of-the-art neural machine translation systems and tested on both in-domain and out-of-domain test sets of large size.

Chapter 3

Cleaning a(n almost) Clean Institutional Translation Memory

Contribution

Shivendra Bhardwaj [1] (first author), David Alfonso-Hermelo[1], Philippe Langlais[1], Michel Simard [2], Cyril Goutte[2] and Gabriel Bernier-Colborne[2]. Cleaning a(n almost) Clean Institutional Translation Memory, 2020. Submitted at 28th International Conference on Computational Linguistics (COLING'2020).

{shivendra.bhardwaj, david.alfonso.hermelo, philippe.iro}@umontreal.ca
{Michel.Simard, Cyril.Goutte, Gabriel.Bernier-Colborne}@nrc-cnrc.gc.ca

In this article, I proposed and conducted the deep-learning experiments, and David designed the heuristics and annotated the data. Professor Philippe Langlais and I worked on the analysis and the writing. The NRC-CNRC team assisted in proofreading and carried constructive discussions.

Abstract

While recent studies have been dedicated to cleaning very noisy parallel corpora for the sake of better Machine Translation, we focus in this work on filtering a mostly clean institutional Translation Memory for the sake of a better internal usage of the memory. This problem of practical interest has not received much consideration from the community. We were provided access to the translation memory of a large institutional translation service, which is extensive and multi-domain. We propose two ways of evaluating this task, manual annotation and Machine Translation, and compare five approaches involving solutions based on heuristics, feature engineering, and deep learning. We report significant gains over a state-of-the-art, off-the-shelf Translation Memory cleaning system.

3.1. Introduction

Over the past few decades, the translation memory (TM) has become a critical component for most translation services. In fact, it can be argued that, apart from human expertise, the TM is a translation service's most valuable asset. Newly received documents are systematically matched against their contents

[1] Researchers from RALI
[2] Researchers from NRC-CNRC

for reuse, reducing cost and increasing consistency for clients. In many organizations, the TM has also become the primary source of information for language professionals, who query them directly through their "concordance" functionality [Bundgaard and Christensen, 2019, Teixeira and O'Brien, 2017]. More recently, TM contents have also been put to use as training data for domain- or client-specific machine translation (MT) systems.

As time passes and translation memories grow in size, their intrinsic value for the organization naturally increases. With this growth, however, also comes an increase in the amount of content that is unfit for reuse. This content, refered to as "noise", "dirt", or even "weed" [Simard, 2014, Young et al., 2016], comes from different places: some is the result of translation-specific issues such as missing or untranslated text in target documents, or other typical translation errors such as the use of calques, improper wordings, etc. Noise can also arise from spelling and morphosyntax errors, or use of terminology and phrasing that does not meet established norms, either in the source or the target versions of the texts. Another important class of problems are those that result from the machinery used to populate the translation memory. Most TMs store text in the form of "translation units" that mostly correspond to sentences, and therefore rely on automatic extraction and segmentation of the text into such units. Improper segmentation can lead to pairs that don't correspond to logical units, making them less usable. In some settings, TMs also rely on automatic alignment methods to pair up the segments from the source and target documents. Again, faulty alignments can render some pairs, and sometimes whole documents, unusable.

TM noise can have severe consequences. Obviously, it can drastically reduce the potential benefits of segment reuse in new documents. But it can also be a major irritant for language professionals, who may come to feel that they can no longer trust the resource. And while statistical MT systems were known to be resilient to noise [Goutte et al., 2012], the situation is quite different with NMT [Khayrallah and Koehn, 2018]. This situation motivates the need for methods for "cleaning" TMs.

In the following pages, we present our work to develop such cleaning methods for the TM of a large institutional translation service. We developed three different methods: a combination of 13 manually-devised heuristics; a supervised classifier based on these heuristics; and a semantic similarity method based on LASER [Artetxe and Schwenk, 2019b]. We compared these to state-of-the-art methods, namely those of Jalili Sabet et al. [2016b] (TMOP) and of Grégoire and Langlais [2018]. Of these five methods, our LASER-based method was the most accurate for identifying good pairs of segments. But when training MT systems on clean subsets of the TM, best results were obtained by combining all five methods.

3.2. Related Work

TM cleaning is most naturally framed as an "error detection" problem: given a set of translation pairs, filter out those that match known patterns of error. In what is possibly the earliest work along this line, Macklovitch [1994] proposes simple heuristics to identify specific problems observed in real (professional) translations, such as errors in numerical entities, the presence of calques, and abnormal translation sizes. The work of Barbu [2015] can be seen as an extension of this line of work: the author proposes 17 features, some based on formal clues (e.g. the presence/absence of XML tags, emails, URLs, numbers, capital letters or punctuation) and others using external resources (i.e. the Bing translation API and the language detector Cybozu). Based on these features, classifiers are trained to recognize bad translations, using a very small training set (1243 sentence pairs). The best model (an SVM model) achieves an F-score of 81% on a test set of 309 sentence pairs. However, the author concludes that applying it on MyMemory [Trombetti, 2009] would filter out too many good sentence pairs.

Alternatively, TM cleaning can be viewed as a variant of parallel corpus extraction, in which the focus is instead on finding "good" translations. A widely-used method for identifying translated sentence pairs (SPs) in a comparable corpus is proposed by Munteanu and Marcu [2005b]. It relies on a feature-based classifier trained in a supervised way. Different features are exploited including length ratio of the source and the target sentences, bilingual lexicon matches, and a set of features based on IBM word translation models [Brown et al., 1993a]. The authors show that the parallel material mined from news extracted over the web improves a downstream statistical translation engine.

More recently, the success of deep learning methods has led to parallel corpus extraction methods trained without feature engineering. Notably, Grégoire and Langlais [2018] describe a siamese recurrent neural network that encodes source and target sentences into vectors that are then fed through a non-linear transformation in order to classify a sentence pair as parallel or not. The authors showed that training such a model yielded better performance than the aforementioned approach, and that adding parallel material extracted from Wikipedia using this model leads to systematic (although modest) gains in both statistical and neural machine translation engines.

However, parallel corpus extraction methods tend to focus more on precision than on recall: find the *really* good pairs rather than find *all* the good pairs. More recent approaches to TM cleaning typically rely on both error detection and corpus extraction methods to reach a better balance. For example, Jalili Sabet et al. [2016b] introduces a fully unsupervised TM cleaning tool called TMOP, which relies on 25 different features, some adapted from Barbu [2015], others based on de Souza et al. [2014], to estimate the quality of the translations. This method also makes use of multilingual word embeddings, using a method proposed by Søgaard et al. [2015]. Each feature acts as a filter and returns a score, which TMOP combines into a final decision. Experiments on a subset of the English-Italian MyMemory produced results comparable to [Barbu, 2015], but without the need for labeled data.

Recently, there has been increased interest in filtering very noisy (web-mined) parallel corpora using unsupervised deep learning. Chaudhary et al. [2019] train multilingual sentence embeddings using LASER and exploit an ensemble of evaluation methods like Zipporah [Xu and Koehn, 2017], Bicleaner [Sánchez-Cartagena et al., 2018], and dual conditional cross-entropy filtering [Junczys-Dowmunt, 2018] to score each sentence pair. Wang et al. [2018] propose a data selection method for de-noising training material and adapting to a specific domain. This kind of approach is less suited to a situation such as ours, however, in which we are dealing with over 200 domains.

Corpus	#SPs	#French types	#English types
MDTM	139.5M	1.1M	1.4M
MT-TRAIN	14.0M	387 594	465 603
MT-TEST	10k	10 733	12 884
META-H	18.9M	570 896	680 603
BALANCED	7.0M	462 703	444 305
2021	2021	3 480	4 304

Tableau 3.1. Characteristics of the corpora used. We count as types space-separated strings that contain only alphabetical symbols and are no longer than 15 characters. The presence of many token types is due among other things to a large proportion of proper names, as well as issues with the many file formats used to store source and translated documents.

3.3. Datasets

The dataset we work with was provided to us by a large Institutional Translation Service, ITS for short. We call this dataset the *Multi-domain translation memory* or MDTM. To conduct our experiments, we created a number of subsets of the MDTM, which we describe below. Their characteristics are summarized in Table 3.1.

The MDTM comprises over 1.8M pairs of documents in English and in French, each encoded as a separate TMX file. Documents are organized into more than 200 broad domains (e.g. health, environment, finance, etc.), totalling over 139 million sentence pairs. In the translation memory system used by the ITS, translators who encounter problematic sentence pairs (“noise”) may flag them, in which case the whole document containing the sentence is labeled as problematic and excluded from future searches. This flagged material represents 7.7% of all sentence pairs. Note that we do not know which specific sentence pair(s) within each flagged document were found to be problematic, only that flagged documents contain at least one such pair.

For training our translation engines (see Section 3.5) we sampled 14M sentence pairs from the MDTM at random so as to obtain comparable amounts of SPs from each domain. 4.3M sentence pairs were sampled from the flagged part of the corpus, so that we could monitor if and how this material impacts MT quality. We call this corpus MT-TRAIN. We also sampled a test set from the MDTM, but excluding the flagged corpus and the bad SPs identified by heuristics (see Section 3.4.1). This was meant to ensure that the test material is not corrupted. We ended up with a subset of 10 000 sentence pairs which we call MT-TEST.

We applied the two meta-heuristics described in Section 3.4.1 to the MDTM: this labeled 17.7M SPs as good, and 1.2M as bad. Those two subsets constitute an annotated corpus we call META-H. Because of the nature of the heuristics we deployed, we observed that many of the bad SPs feature obvious errors (presence of gibberish, typos, non-translations, etc.), while there are fewer misaligned SPs and SPs involving subtle translation errors.
We therefore enhanced META-H automatically by: a) taking existing English sentences and pairing them at random with existing French sentences, for a total of 1.15M artificially created misaligned SPs, and b)

replacing some (source or target) tokens containing four or more characters with one of their top five nearest neighbours in a space of fastText word embeddings [Bojanowski et al., 2017]. We produced 1.15M sentence pairs with (often not so) subtle problems, such as those in Figure 3.1, where nearest neighbours of a word are often typos. After this addition, we obtained 3.5 million bad SPs. To create a balanced set of 7 million SPs, we combined these with 3.5M pairs selected randomly among those identified as good. This balanced corpus named BALANCED is used to train our supervised classifiers (in Section 4.5).

ori	If you feel like sleeping , stand up and move to back .	ori	The government of Canada will match your contribution dollar for dollar .
cor	If you feels just napping, Stand up and moves to abck .	cor	The governnment of Quebec will match your Contribution dolllar for dollar.

Figure 3.1. Examples of original (ori) and corrupted (cor) sentences.

Finally, in the course of our experiments, we conducted targeted manual evaluations that we compiled into a corpus named 2021, which we use for evaluation purposes. This corpus includes the 1721 SPs used to adjust the meta-heuristics and the 300 SPs used to evaluate them, as explained in Section 3.4.1. We found 1182 (58.5%) good SPs and 839 (41.5%) bad SPs, making 2021 a rather balanced corpus.

3.4. Approaches

We compared five different approaches to identify noisy sentence pairs, three of them being unsupervised (pre-trained or not trained at all).

3.4.1. Heuristics

By inspecting the MDTM, we noticed a number of problems, some of which (we thought) could be detected by specific rules, as illustrated by the examples in Figure 4.2. We developed 13 heuristics, which we detail below. Each takes as input a sentence pair (SP) and produces a score between 0 (noisy SP) and 1 (good SP). Most of those heuristics are exploited in one way or another in the systems described in Section 3.4.2.

A first set of heuristics looks for matches (i.e. either exact string matches or bilingual matches based on a lexicon) between the two sentences, each looking for specific units such as numerical entities (NUM), cognates (COG), stop words[3] (STOP), URLs (URL), false friends[4] (FRIEND), punctuation and specific symbols (PUNC). Another heuristic (LEX) counts the number of word translation pairs found according to a bilingual lexicon containing 60k entries. Another heuristic (ION) takes into account the fact that in languages whose vocabulary contains many latinate words, words ending with the suffix -ion are often translated by words with the same suffix (e.g. félicitations / congratulations).

A second set of heuristics detects specific problems such as the presence of gibberish (GIBB), which we found abundant, or the presence of a source sentence in place of a target one (MONO), most often because parts of a text were not translated, but not filtered out by the pipeline that feeds the MDTM. We also noticed

[3] We use a lexicon of 93 entries such as the / la, le, les.

[4] We use a lexicon of 175 entries such as fabric / fabrique.

1) en Section 34 verification and certification
fr Fiches de spécimen de signature.

2) en Native Women's Association of Canada.
fr James Anaya, Doc. NU A/HRC/9/9, 11 août 2008. Native Women's Association of Canada.

3) en Since 2005, we have received some $1.4 billion to purchase 17 vessels.
fr Conflicting sovereignty claims to the Arctic are resulting in a race to the North.

4) en OP #L O- Sk
fr i@n [u05ce \x9b}

Figure 3.2. Examples of typical problems in the MDTM: 1) poorly aligned section titles; 2) partial misalignment, involving sentence segmentation problems and mixed languages in the French version; 3) "French" version in English; 4) character encoding problem, possibly the result of the multiplicity of formats handled by the ITS.

many problems involving tables of contents (TOC), which often confuses the sentence segmenter, leading to alignments errors. We also check the length ratio (counted in words) of source and target sentences (LEN). Finally, we implemented a rudimentary proxy to spell checking (SPELL), which counts the number of tokens that are correctly spelled (according to a list of words seen at least 1000 times in Wikipedia).

In order to gauge the performance of each heuristic, we manually annotated 1721 SPs: We randomly sampled 1321 SPs from the MDTM, to which we added 400 sentence pairs that had specific problems (e.g. unbalanced number of cognates). We annotated the resulting 1721 sentence pairs as good or problematic. We used this corpus to adjust the thresholds of our heuristics and select the optimal combination of heuristics to distinguish good and bad sentence pairs.

For detecting good SPs, the solution that showed the best precision results (on a test set of 300 manually annotated SPs) is a weighted combination of 4 heuristics: NUM, LEX, ION, and PUNC. In contrast, the combination with the best precision (on the aforementioned test set) to predict problematic SPs is a combination of 9 heuristics: LEN, MONO, GIBB, NUM, SPELL, URL, LEX, TOC and PUNC. To further evaluate these two "meta-heuristics" (i.e. weighted combinations of heuristics), we manually inspected random samples of 150 SPs selected by each (i.e. identified as good or bad respectively). One annotator found 115 good SPs in the former sample (76.7%), and 121 bad SPs in the latter (80.7%).

3.4.2. Feature-based Classifiers

Based on the heuristics of Section 3.4.1 we trained classifiers on BALANCED to identify good SPs. In addition to the score (between 0 and 1) produced by each of the 13 heuristics, we included as features intermediate values produced while computing the heuristics, for a total of 60 scores. To that, we added two aggregate features: the percentages of heuristics that identify an SP as good (resp. bad). We trained both support vector machines (SVM) and random forests (RF) on the BALANCED training set, using the aforementioned features and utilizing `scikit-learn`[5] on standard desktop CPUs. Training took approximately 10 hours per model.

3.4.3. TMOP

TMOP [Jalili Sabet et al., 2016b] is composed of 25 different binary functions meant to capture bad alignments, bad translation quality or "semantic" distance between the source and target. It offers 3 ready-made configurations which control how those functions are aggregated into a final decision. We used the one which classifies an SP as bad if at least 5 functions signal a problem, and good otherwise.[6]

Similarly to [Munteanu and Marcu, 2005b], TMOP relies (among other things) on IBM-like features computed through the MGIZA++ package [Gao and Vogel, 2008b].[7] It took 13 days to run MGIZA++ through the 14M SPs of the MT-TRAIN corpus on a 16-core computer equipped with 70Gb of memory, and 5 more days to run TMOP on it.[8] Therefore, applying TMOP on the full MDTM would be rather challenging.

3.4.4. Deep-Learning Classifier

We reimplemented in Keras [Chollet et al., 2015] the model of Grégoire and Langlais [2018], introducing a few variants we found useful. The model architecture consists of two bidirectional LSTMs [Hochreiter and Schmidhuber, 1997a], each with 300 hidden units[9] which encode sentences into two continuous vector representations.

The source and target representations are then fed into a Feed-Forward Neural Network containing two hidden layers (with 150 and 75 units respectively), followed by a sigmoid activation function, which outputs the probability that the SP is good.

We trained our model using the Adadelta optimizer [Zeiler, 2012] with gradient clipping (clipped at 5) to avoid exploding gradient and a batch of size 300 (whereas the original implementation uses the Adam optimizer with a learning rate of 0.0002 and a mini-batch of 128).[10] Models were trained using 4 Tesla V100-SXM2 for 10 epochs, which took approximately 2.5 hours.

3.4.5. LASER

We also devised a simple cleaning method based on the LASER [Artetxe and Schwenk, 2019b] toolkit.[11] The idea behind this method is to train a single encoder to handle multiple languages such that semantically

[5] `https://scikit-learn.org/stable/`

[6] The "twenty percent" configuration was by far the better: the "one reject" configuration is producing far too many false negatives, while the "majority vote" configuration hardly detects bad SPs.

[7] `https://github.com/moses-smt/mgiza`

[8] We had to run TMOP on a dedicated cluster of 32 CPUs equipped with 300Gb of memory. Training embeddings took only 6 hours of the 5 days required in total.

[9] In the original paper, the authors use 512-dimensional word embeddings and 512-dimensional recurrent states since they learn the word embeddings from scratch. We found it easier and faster to adapt pre-trained, 300-dimensional `fastText` word embbedings. Also, the authors tie the parameters of the two encoders, while we do not.

[10] Adadelta does not require us to set a default learning rate, since it takes the ratio of the running average of the previous time-steps to the current gradient.

[11] `https://github.com/facebookresearch/LASER`.

similar sentences in different languages are close to one another in the embedding space. We used a pre-trained sentence encoder that handles 92 different languages. Sentences from all these languages were mapped to a common embedding space using a 512-dimensional bi-LSTM encoder.

For each source-language sentence s_i at index i, we find the target language sentence t_j that is closest in the the joint embedding space, using the `multilingual-similarity search` (MSS) method provided with the toolkit. If both sentences have the same index (i.e. $i = j$), the sentence pair is considered good, otherwise not. There is no training involved in this process, since we consume the model as it is. Running this method on one Tesla V100-SXM2 took approximately 14 hours.

3.5. Neural Machine Translation Models

Evaluating the quality of a translation memory by the performance of a translation engine trained on that memory is rather intuitive. Still, many factors can render this methodology troublesome. In order to draw conclusions that are independent of a specific system, we experimented with two very different neural translation models: XLM, a deep transformer model, and ConvS2S, a convolutional seq2seq model. Both models were trained in parallel on 4 Tesla V100-SXM2. The average time to train XLM was around 22-30 hours depending on the data set, and for ConvS2S, it was 72-96 hours.

3.5.1. Cross-lingual Language Model

In [Conneau and Lample, 2019a], the authors propose three models: two unsupervised ones that do not use sentence pairs in translation relation, and a supervised one that does. We focus on the third model, named Translation Language Modeling (TLM) which tackles cross-lingual pre-training in a way similar to the BERT model [Devlin et al., 2018b] with notable differences. First, XLM is based on a shared source-target vocabulary of sub-words, computed using byte pair encoding (BPE) [Sennrich et al., 2016]. We used the 60k BPE vocabulary which comes with the pre-trained language model.[12] Second, XLM is trained to predict both source and target masked words, leveraging both the surrounding words and the other language context, encouraging the model to align the source and target representations. Third, XLM embeds the language of the tokens as well as their position in the sentence, which leads to build a relationship between the related tokens in the two languages.

XLM is implemented in PyTorch and supports distributed training on multiple GPUs. We have modified the original pre-processing code so that XLM can accept a parallel corpus for training TLM.[13] The translation is produced by a beam search strategy, making use of a beam width of 6 and a unity length penalty.

3.5.2. Convolutional Sequence to Sequence

The section predominant method to sequence to sequence (seq2seq) learning is to map an input sequence to an output sequence of variable length via a recurrent neural network, e.g. an LSTM. The work of Gehring et al. [2017] demonstrated that convolutional neural networks (CNN) could also be used for seq2seq. The

[12] Training TLM without pre-training was rather unstable. We also noticed better results with a back-translation step, but at a high cost in training time.

[13] At the time of this writing, this was not implemented in the XLM GitHub repo (`https://github.com/facebookresearch/XLM.git`).

Method	(Section)	Accuracy (%)
meta-heuristics	4.1	42
SVM	4.2	63
RF	4.2	60
TMOP	4.3	60
bi-LSTM	4.4	79
LASER	4.5	**84**

Tableau 3.2. Accuracy of detection of good vs. bad SPs on 2021, for various approaches. Supervised methods are trained on BALANCED. The first line indicates the score of the meta-heuristics, see Sec. 3.4.1.

ConvS2S model uses CNNs with Gated Linear Units [Dauphin et al., 2017] for both the encoder and decoder, and includes a multi-step attention layer.

We used the implementation available in the fairseq toolkit [Ott et al., 2019]. Similarly to TLM, we use a source and target vocabulary of 60K BPE types. The translation is generated by a beam-search decoder with log-likelihood scores normalized by sentence length.

3.6. Experiments

3.6.1. Manual Evaluation

Table 3.2 shows the accuracy on detecting clean vs. incorrect SPs, computed on the 2021 corpus, for the classifiers we trained on the BALANCED corpus, as well as the unsupervised systems. We observe that training a classifier (SVM, RF) on top of the features used by meta-heuristics clearly helps. We also observe that TMOP delivers comparable results to our heuristic-infused classifiers, which indicates that it is a good detector on its own, and that combining heuristic signals to word-based translation and embedding features is a good strategy. Further learning how to aggregate those signals would likely improve the performance.

The bi-LSTM model simultaneously avoids feature engineering and leads to much better results overall, which confirms the observations made by Grégoire and Langlais [2018] on artificial data. What comes as a surprise is that the best results are obtained by LASER, and this, without any additional training on our data. Of course, 2021 is a rather small test set, and results here should be taken with a grain of salt.

3.6.2. Machine Translation Evaluation

Table 3.3 shows the BLEU scores obtained by the different neural translation engines we trained. Removing the flagged material from the training set (see line 2) does not impact BLEU scores significantly, which corroborates the observations we made that the flagged material was generally of good quality. We observe (line 3) that TMOP, while providing a small improvement over TM-TRAIN, has problems identifying bad SPs, and is therefore the worst filtering strategy. Clearly, some adaptation to our dataset would in practice be required if we had to deploy it efficiently on MDTM.

From Table 3.3, we can observe that cleaning the training material by any of the methods described leads to some gains in BLEU. We were not expecting such a clear outcome, with a corpus as clean as the MDTM. The largest gains come from the bi-LSTM approach, a supervised approach, but LASER is not far behind, even though it doesn't fare as well as in the manual evaluation (Section 3.6.1). The former approach filters much more, which seems to improve MT performance further. It should be noted that we applied the supervised filters on the non-flagged part (line 2), while the LASER unsupervised method was applied directly to MT-TRAIN.

The differences in BLEU obtained by the two feature-based classifiers do not differ much, with SVM being at a slight edge in the manual evaluation. Lastly, we observe that the gains are consistent over the two translation engines, which is reassuring.

Train set	**#SPs (millions)**	XLM	ConvS2S
MT-TRAIN	14.00	36.25	33.04
¬Flagged	9.67	36.29	33.33
TMOP	13.38	36.49	33.51
RF	7.20	36.70	33.72
SVM	7.50	36.53	33.91
meta-H	8.15	36.80	33.78
LASER	9.65	37.23	33.58
bi-LSTM	6.13	37.52	**33.96**
∩ALL	5.80	**37.57**	33.93
random	5.80	36.31	33.00

Tableau 3.3. BLEU scores of the XLM and ConvS2S translation engines. Corpus size (#SPs) is in millions of sentencepairs. ¬Flagged is the part of MT-TRAIN that has not been flagged by a professional translator (see Section 3.3).

In the left part of Figure 3.3, we plot the BLEU scores obtained on the test set over epochs by the different XLM translation engines we trained. We observe similar training curves (including a notable increase of performance at epoch 10) for all systems, and that filtering the MDTM leads to gains at each epoch.

3.6.3. Analysis

Our manual evaluation, although conducted on a limited number of SPs (2021), demonstrates that our cleaning approaches are effective. This is also confirmed by BLEU improvements. The nature of the cleaning performed is however not clear. We investigate this issue in the following.

3.6.3.1. *Domain Specificity*

To avoid domain adaptation of our models, we make sure that the distribution of domains (client names) in MDTM (P) have a close KL divergence (presented hereafter) with MT-TRAIN (Q) and its filtered versions (KL=0.0110):

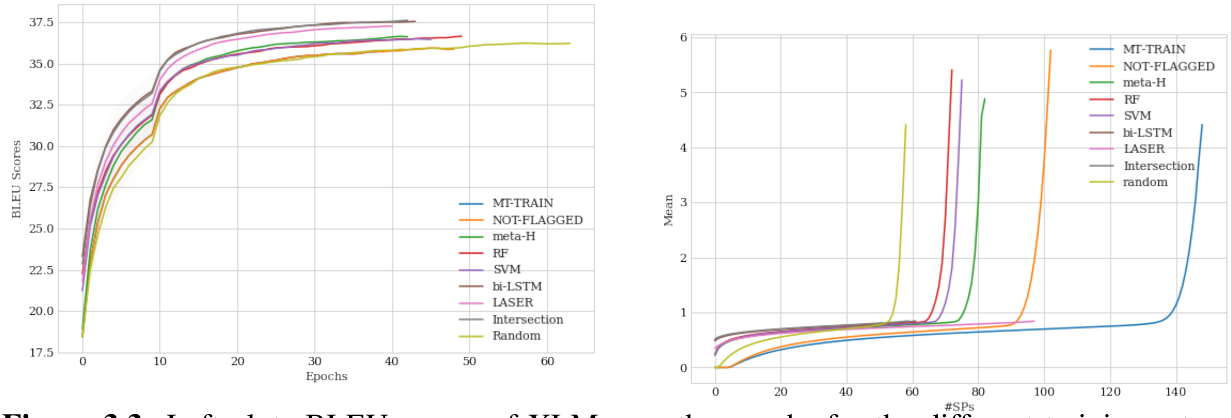


Figure 3.3. Left plot: BLEU scores of XLM over the epochs for the different training sets we considered. Right plot: average length-ratio of sentence pairs of the different sub-corpora of MT-TRAIN as a function of the number of slices of 100k SPs considered (see Section 3.6.3.4)

$$\mathbf{D}_{KL}(P||Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)}$$

3.6.3.2. *Combining Methods*

We observed that the different filtering methods we tested remove different portions of MT-TRAIN. We measured the percentage of sentence pairs that passed the meta-H, bi-LSTM and LASER filters and observed that 93.4% of SPs detected by LASER as bad were also detected by bi-LSTM, while the agreement on bad SPs between LASER and either META-H or RF is around 65%. If we remove SPs detected bad by these three methods, we end up with a subset of MT-TRAIN which contains 5.8M sentence pairs, the performance of which is reported on line ∩ALL of Table 3.3. BLEU scores are not significantly different from those obtained with bi-LSTM alone, which tends to indicate that the deep supervised classifier does not benefit from other methods. As a sanity check, we randomly selected from MT-TRAIN a subset with the very same number of SPs and observed a drop in BLEU score of -1.26 and -0.93 for XLM and ConvS2S respectively (see the last line of Table 3.3). This supports the claim that our methods do perform cleaning of the translation memory.

3.6.3.3. *Unknown Words*

Because both translation engines are using a fixed set of 60K BPE subword units, some target words in the training material can not be reproduced by the systems. For MT-TRAIN, we measure over 76k such token types for the XLM translation engine. Filtering the training material with feature-based classifiers, bi-LSTM and LASER, lowers this rate to around 3k, 450 and 17k token types respectively, a clear reduction, especially with bi-LSTM. By inspecting why it was so, we observed that the vast majority of types that could not be reproduced by concatenating BPE units were gibberish words (e.g. t0DÉlmsˋaoyCœ). This indicates that our cleaning approaches are taking care of this problem.

3.6.3.4. *Length-ratio*

We report in the right plot of Figure 3.3 the length-ratio of sentence pairs in the different sub-corpora identified from MT-TRAIN. Length is counted in words, and we report the average length-ratio (|en|/|fr|) computed on increasing slices of 100k sentence pairs. Prior to computing averages, SPs in a corpus are sorted in increasing order of length-ratio. The (blue) curve is the distribution obtained on the MT-TRAIN corpus. It starts with near zero mean (much longer French sentences), then gradually increases to a ratio slightly lower than one (English sentences are typically shorter than French ones), and finally peaks at around 4, the average over the full corpus. Near zero and high ratio very likely indicate alignment problems. We observe that applying meta-heuristics or feature-based classifiers reduces near-zero and high ratios to some extent. It is much more noticeable that bi-LSTM and LASER remove most of them, leading to an average length-ratio of around 0.8. This observation further supports the fact that cleaning is being performed.

3.6.3.5. *Other Evidence of Cleaning*

We computed the ratio of sentence pairs in which a `www` token is in the English sentence but not in the French part. We have 25 203 such sentence pairs in MT-TRAIN, over 57 925 pairs with `www` tokens that do match, a ratio of 43.5%. Feature-based classifiers reduce this ratio to around 10.5%, while bi-LSTM and LASER lower the ratio to 3.1% and 4.9% respectively.

Finally, we manually annotated 100 sentence pairs belonging to the ∩ALL set in Table 3.3, that is, SPs classified good by all approaches, as well as 100 sentence pairs classified bad by both deep learning methods. We found 16 false positives (SPs wrongly identified as good) in the former set, and 33 false negatives (SPs wrongly identified as bad) in the latter set. This last figure suggests that deep learning methods are too strict noise detectors.

3.7. Conclusions

In this work, we report on our path to filtering a mostly clean Translation Memory for professional use. Our experiments show that, among the different methods tested, the pre-trained LASER and the in-house trained bi-LSTM are able to discriminate bad from good sentence pairs with high accuracy. The former approach is unsupervised and delivers the best results on our small scale manual evaluation. These two methods outperform heuristics we devised specifically for this task, as well as feature-based classifiers we trained on datasets selected using these heuristics. It also clearly outperform the TMOP system which turned out to be very challenging to deploy.

By filtering a large training set using our methods, we obtain machine translation gains. Similarly to our manual evaluation, deep learning methods lead to larger gains in BLEU. The bi-LSTM classifier could remove over half of the training material while improving BLEU by over one point. In future research, we would like to revisit a number of choices we made. For instance, when we created a balanced corpus of (supposedly) good and bad sentence pairs, we were not very successful in generating artificial SPs with subtle errors. Also, we overlooked the nature of noise we are trying to identify. It is for instance currently difficult to ascertain whether we are able to identify subtle errors (such as bad wordings), an avenue which deserves more investigation.

Chapter 4

Human or Neural Translation?

Contribution

Shivendra Bhardwaj [1] (first author), David Alfonso-Hermelo[1], Philippe Langlais[1], Michel Simard [2], Cyril Goutte[2] and Gabriel Bernier-Colborne[2]. Human or Neural Translation?, 2020. Submitted at 28th International Conference on Computational Linguistics (COLING'2020).
{shivendra.bhardwaj, david.alfonso.hermelo, philippe.iro}@umontreal.ca
{Michel.Simard, Cyril.Goutte, Gabriel.Bernier-Colborne}@nrc-cnrc.gc.ca

In this paper, I designed and conducted all the experiments and Professor Philippe Langlais carried out the analysis section, and also assisted me in the writing. David and the NRC-CNRC team helped in proofreading.

Abstract

Deep neural models tremendously improved machine translation. In this context, we investigate whether distinguishing machine from human translations is still feasible. We trained and applied 18 classifiers under two settings: a monolingual task, in which the classifier only looks at the translation; and a bilingual task, in which the source text is also taken into consideration. We report on extensive experiments involving 4 neural MT systems (Google Translate, DeepL, as well as two systems we trained) and varying the domain of texts. We show that the bilingual task is the easiest one and that transfer-based deep-learning classifiers perform best, with mean accuracies around 85% in-domain and 75% out-of-domain.

4.1. Introduction

This work addresses the task of distinguishing between translations produced by humans and machines. Practical applications for this include: improving machine translation systems [Li et al., 2015a], filtering parallel data mined from the Web [Arase and Zhou, 2013] and evaluating machine translation quality without reference translations [Aharoni et al., 2014]. In our case, we are more interested in tracing the

[1] Researchers from RALI
[2] Researchers from NRC-CNRC

origin of translations outsourced by a large institutional translation service.

Our work aims at distinguishing between human and neural machine translations at the sentence level. We consider two settings: a monolingual task, where only the target sentence is considered; and a bilingual task where both the source text and its translation are available. We compare feature-based approaches with several deep learning methods, investigating the impact of text domains and MT systems (in-house neural engines, Google Translate, DeepL), paying attention to cases where the translation engine at test time is different from the one used for training, which we found often not studied in related work.

We show that identifying machine translation is still feasible nowadays. On the bilingual task, the best transfer learning method we tested recorded an in-domain accuracy of 87.6% and out-of-domain performances ranging between 65.4% and 84.2% depending on the domain of texts and MT system considered. We analyze why our classifiers manage to do better than chance even though translations produced automatically seem to us of very good quality overall. We believe our study offers many new data points, and hope it will foster research on this timely topic.

After reviewing related work in Section 4.2, we describe our dataset and experimental setting in Section 4.3, the neural MT systems we used in our experiments in Section 4.4 and the classifiers we tested in Section 4.5. We present our experimental results in Section 4.6 and propose a deeper analysis in Section 4.7.

4.2. Related Work

Most studies on identifying machine translation were conducted at a time where MT systems were fraught with problems that rendered their identification somewhat easy. Current neural MT systems deliver translations that are sometimes bafflingly fluent. We are not aware of much work addressing MT identification with these newer systems. One notable exception is a recent study by Nguyen-Son et al. [2019c] on distinguishing original sentences from translations produced by Google Translate (GT). The authors build on the interesting intuition that back translations of translated texts should be less modified than back translations of original (not translated) ones. They report an accuracy of 75% with an SVM classifier on a corpus of 1200 sentences that are either original (not translated) sentences or translated with GT.

In earlier work on MT identification, approaches and evaluations vary greatly from one study to the other. For instance, Li et al. [2015a] uses features extracted from the parse tree of the sentence to characterize, as well as features capturing the density of some function words (with the help of a part-of-speech tagger), and some features dedicated to out-of-vocabulary words. They also use features aimed at capturing emotion agreement inside a sentence, using a dictionary of emotion words. They gathered a balanced dataset of human and machine translations from the Europarl corpus[3] using a statistical machine translation (SMT) engine trained in-domain with Moses [Koehn et al., 2007]. They report an accuracy of 74.2%. However,

[3] http://www.statmt.org/europarl

they do not analyze which features are the most beneficial to the task.

Arase and Zhou [2013] investigate the use of features to capture the fluency of the text, such as part-of-speech and word-based n-gram language models, as well as features aimed at detecting so-called *phrase-salad* phenomena [Lopez, 2008], i.e. poor inter-phrasal coherence often observed in SMT output. On a collection of public texts crawled over the Web, they report an accuracy of 95.8% when distinguishing human versus automatic translations. The best performance was observed when combining all the features, and surpasses that of humans performing the same task (88.2%).

Aharoni et al. [2014] use features capturing the presence or absence of part-of-speech tags and function words taken from LIWC [Pennebaker et al., 2001] appearing at least 10 times in the training material. On a corpus extracted from the Canadian Hansards, and using various translation engines, they report accuracies at detecting machine versus human translations which are inversely correlated with the quality of the MT system used. For the best systems, they report an accuracy slightly over 60%.

4.3. Data

All our experiments are centered around one very large dataset: the translation memory of a large institutional translation service. This data collection — called TM hereafter — contains the English and French versions of over 1.8 million documents, covering over 200 broad domains (military, health, etc.), for a total close to 140 million sentence pairs. Since the vast majority of translations in the TM are into French, we focus on this language direction.

Our goal is to build classifiers that determine if a translation is human or machine-made. For this, we need training data that contains both types of translations. We create such data by machine translating a subset of 530k sentence pairs, randomly sampled from the TM. These machine translations are performed using two different neural MT systems, themselves trained using a distinct subset of 5.8M sentence pairs, also randomly sampled from the TM.[4] These two MT systems, one based on XLM [Conneau and Lample, 2019b] and one on FairSeq [Ott et al., 2018a], are detailed in Section 4.4. Thus, two distinct classifier training sets are created, one from each MT system: each contains 530k human translations and 530k machine translations, totalling 1.06M examples.

We proceed similarly to produce test sets to evaluate the performance of our classifiers: we randomly sample 10k sentence pairs from the TM, machine translate the English versions into French using our XLM and FairSeq MT systems, thus creating two test sets of 20k examples (10k human translations + 10k machine translations) each. We call these X-TM (for XLM) and F-TM (for FairSeq).

These two test sets can be seen as "in-domain" relative to our classifiers: not only because they share the same source as the training data (the TM), but also because the machine translations were produced using the same MT systems. To test the ability of our classifiers to handle different text domains and

[4] All sampling in the TM was done in such a way as to ensure comparable representations of each domain.

translations produced by different MT engines, we also created "out-of-domain" test sets: we used two online translation platforms — DeepL[5] (D) and Google Translate[6] (GT) — to translate 10K sentences of each of four publicly available data sets: Europarl (EURO), Canadian Hansard (HANS),[7] the News Commentaries (NEWS) available through the WMT conference,[8] and the Common Crawl corpus (CRAWL) also available through WMT. Again these were mixed in equal parts with human translations. In what follows, each test set is named based on the system used to produce automatic translations, and the domain of the material.

We further translated another excerpt of (previously unused) 10k sentences from the TM, using the *DeepL* translation API with a private account, to produce a test set we call D-TM. The TM being a proprietary translation memory, we did not submit it to the GT platform.

4.4. NMT systems

As noted above, to produce the training data for our classifiers, we first created two transformer-based NMT systems using English-French texts from the TM. We provide the details of this process here.

4.4.1. Cross-lingual Language Model (XLM)

In [Conneau and Lample, 2019b], the authors propose three models: two unsupervised ones that do not use sentence pairs in translation relation, and a supervised one that does. We focus on the third model, called the Translation Language Modeling (TLM) which tackles cross-lingual pre-training in a way similar to the BERT model [Devlin et al., 2019] with notable differences. First, XLM is based on a shared source-target vocabulary using Byte Pair Encoding (BPE) [Sennrich et al., 2016]. We used the 60k BPE vocabulary which comes with the pre-trained language model.[9] Second, XLM is trained to predict both source and target masked words, leveraging both source and target contexts, encouraging the model to align the source and target representations. Third, XLM stores the ID for the language and the token order (*i.e.*, positional encoding) in both languages which builds a relationship between related tokens in the two languages.

During training and when translating, we use a beam search of width 6 and a length penalty of 1. XLM is implemented in PyTorch[10] and supports distributed training on multiple GPUs.[11] The original distribution does not include beam search for translating (but does for training), so we modified it accordingly. Also, we modified the pre-processing code such that XLM accepts a parallel corpus for training TLM.[12]

[5] `www.deepl.com/translator`

[6] `https://translate.google.com/`

[7] `https://www.isi.edu/natural-language/download/hansard/`

[8] `https://www.statmt.org/wmt14/translation-task.html`

[9] We found the model without pre-training rather unstable, and noticed better results with a back-translation step, but at a too high cost in training time.

[10] `https://pytorch.org/`

[11] `https://github.com/facebookresearch/XLM.git`

[12] Our modifications will be me made available.

4.4.2. Scaling Neural Machine Translation (FairSeq)

Scaling NMT [Ott et al., 2018a] is a novel transformer model that showcased an improvement in training efficiency while maintaining state-of-the-art accuracy by lowering the precision of computations, increasing the batch size and enhancing the learning rate regimen. The architecture uses the `big-transformer` model with 6 blocks in encoder and decoder networks. The half-precision training reduced the training time by 65%. Scaling NMT is implemented in PyTorch and is part of the `fairseq-py` toolkit.[13]

We use the default 40k vocabulary with a shared source and target BPE factorization. During training and for translating, we use a beam search of width 4 and a length penalty of 0.6. For translation,[14] we average the last five checkpoints.

4.4.3. Post-processing

Translating the classifier training data (Section 4.3) with the XLM engine took approximatively 10 hours on a computer equipped with a V100-SXM2 GPU, and 26 hours for the FairSeq system. By inspection, we noticed small issues with the translations produced by both systems, such as punctuation misplacements, extra spaces, inconsistencies in the use of single and double-quotes.

Since those issues would ease the identification of machine-translated material, we normalized the translations in a post-processing step, using 12 very conservative regular expressions[15] that we applied to both the human and machine translations. We observe in Table 4.1 a clear increase of BLEU when applying normalization: +4 for XLM, and +5.3 for FairSeq.

	raw	normalized
XLM	33.43	37.46
FairSeq	34.07	39.40

Tableau 4.1. BLEU scores of the XLM and FairSeq translation engines measured on a dataset of 550K sentence pairs (described in Section 4.3) before (left) and after (right) normalization,

4.5. MT Identification

We experimented with two strategies for building classifiers: feature-based models trained from scratch, as well as deep learning ones making use of pre-trained representations.

4.5.1. Feature-based Classifiers

We considered three supervised classifiers informed by different feature sets. We tested various classifiers (random forest, support vector machines and logistic regression), but obtained more stable results with

[13] `https://github.com/pytorch/fairseq`.

[14] We used the `fairseq-interactive` module of the `fairseq-py` toolkit[13].

[15] Very specific rules such as `replace(' ;',';')` or `replace('https :','https:')`.

random forest classifiers trained with `scikit-learn` [Pedregosa et al., 2011]. In all our experiments, we fixed the number of trees in the forest to 1000 with a maximum depth of 40 and a minimum number of samples required to split an internal node set to 10.

n-GRAM. We reproduce the approach of Cavnar and Trenkle [1994] where we define a vector space on the 30k most frequent character n-grams in the MT output of our training material, with n ranging from 2 to 7.[16] Each sentence is then encoded by the frequency of the terms in this vocabulary, thus leading to a large sparse representation which is passed to a classifier. In the bilingual task, we also consider the top 30k n-grams of the source-language version of the training corpus, leading to representations of 60k dimensions.

KENLM. As a point of comparison, in the monolingual task, we experimented with features extracted from four $\{3{,}4\}$-gram word language models trained with the `kenLM` package [Heafield et al., 2013] on the machine-translated material of our training corpus: two left-to-right models, and two right-to-left ones. We computed 18 features: ratios of min and max `logprob` over the (target) sentence per model (four features), the number of tokens with a `logprob` less than $\{mean, max, -6\}$ (three features per model), as well as the `logprob` of the full sentence given by the left-to-right models (two features).

TMOP. TMOP [Jalili Sabet et al., 2016b] is a translation memory cleaning tool which computes 27 features for detecting spurious sentence pairs, including broad features (such as length ratio) adapted from [Barbu, 2015], some based on IBM models computed by MGIZA++ [Gao and Vogel, 2008a], as well as some features based on multilingual word embeddings, using the method proposed by Søgaard et al. [2015]. While in TMOP, those features are aggregated in an unsupervised way (that is, with rules), we instead pass them to a random forest classifier trained specifically to distinguish human from machine translations. Because of the nature of the feature set, we only deploy this classifier in the bilingual task.

4.5.2. Deep Learning Classifiers

bi-LSTM. We re-implemented the method of Grégoire and Langlais [2018] for recognizing whether two sentences are translations of each other: two bidirectional LSTMs [Hochreiter and Schmidhuber, 1997a] encode the source and target sentences into two continuous vector representations, which are then fed into a Feed-Forward Neural Network with two layers (one in the original paper): one of dimension 150 to process the continuous representation, and one of dimension 75. The output of each network is finally passed to the sigmoid function.

In the original paper, the authors used 512-dimensional word embeddings and 512-dimensional recurrent states since they learn the word embedding from scratch. We found easier (faster, and slightly better) to adapt pre-trained `fastText` word embeddings [Bojanowski et al., 2017] of dimension 300. Also, the authors tie the parameters of the two encoders, while we do not. We use two hidden layers before the sigmoid function because we are mapping from 300 values to 1 and intuitively, it is better to do it smoothly. We trained our classifier with the Adadelta optimizer [Zeiler, 2012] with gradient clipping (clip value 5)

[16] Larger vocabularies do not yield notable performance differences.

to avoid exploding gradient and batch size 300, whereas the original architecture uses the Adam optimizer with a learning rate of 0.0002 and a mini-batch of 128.[17]

We use a similar setting for the monolingual task, except that we only use one bidirectional LSTM whose output we directly pass to the hidden layer of dimension 150, then a layer of dimension 75 and finally the sigmoid function.

LASER. The LASER toolkit [Artetxe and Schwenk, 2019a] released by Facebook[18] provides a pre-trained sentence encoder that handles 92 different languages. Sentences from all the languages are mapped together into the same embedding space with a bi-LSTM 512-dimensional encoder, such that the embeddings from different languages are comparable.

For the bilingual detection task, we extract the representation of the source and target sentences and tie them into one vector by taking their absolute difference and dot product, and adding them. This tied representation is then passed through 3 hidden layers of size 512, 150 and 75 respectively[19] with dropout [Srivastava et al., 2014] of 50%, and then fed into a relu [Nair and Hinton, 2010] activation function, whose output is finally passed to the sigmoid function. For the monolingual task, we just use the LASER French (target) representation of the sentence and pass it through the very same architecture. We train the classifiers with the Adadelta optimizer with gradient clipping (clip value 3).

Transformer-based Classifiers. The use of pre-trained language models in a transfer learning setting is ubiquitous and has shown substantial improvements in various NLP tasks. Therefore, we also considered various representations trained either solely on French data (CamemBERT, FlauBERT) or on multiple languages (XLM-RoBERTa, XLM, and mBERT).

We experiment with different pre-trained transformer models, using the Python module `simpletransformers`[20] based on the `HuggingFace` library[21], which has a sequence classification head on top (a linear layer on top of the pooled output). Our classifiers were fine-tuned using the `ClassificationModel` class and evaluated with the `eval_model` class. We have maintained the same parameters for all the transformer models: sequence length of 256, batch size of 32, Adam optimizer [Kingma and Ba, 2015][22].

CamemBERT: [Martin et al., 2019] is based on the RoBERTa [Joshi et al., 2020] architecture (which is basically a BERT model with improved hyper-parameters for robust performance) and is trained on 138GB of plain French text taken from multilingual corpus OSCAR [Ortiz Suárez et al., 2019].

[17] Adadelta does not require to set a default learning rate, since it takes the ratio of the running average of the previous time-steps to the current gradient.

[18] `https://github.com/facebookresearch/LASER`.

[19] We used three layers here because the input dimension is larger (512 versus 300), but have not investigated the impact of this choice.

[20] `https://github.com/ThilinaRajapakse/simpletransformers`

[21] `https://github.com/huggingface/transformers`

[22] lr: $1 \times e^{-5}$, adam_epsilon: $1 \times e^{-8}$

Unlike RoBERTa, CamemBERT uses sentence piece tokenization [Kudo and Richardson, 2018] and performs whole word masking, which has been shown to be preferable [Joshi et al., 2020]. The architecture of the base model is a multi-layer bidirectional transformer [Devlin et al., 2019, Vaswani et al., 2017] with 12 transformer blocks of hidden size 768 and 12 self attention heads.

FlauBERT: [Le et al., 2020] The base model we used is trained on 71GB of publicly available French data and the data was pre-processed and tokenized using a basic French tokenizer [Koehn et al., 2007]. The model was trained with the MLM training objective.

XLM-RoBERTa: [Ruder et al., 2019] is a multilingual language model, trained on 100 different languages. It is an extended version of XLM (see Section 4.4.1).

mBERT: [Devlin et al., 2019] is very similar to the original BERT model with 12 layers of bidirectional transformers, but released as a single language model trained on 104 separate languages from Wikipedia pages, with a shared word piece vocabulary. The model does not use any marker for input language and the pre-trained model is not made to extract translation pairs to have similar representations. The tokenization splits words into multiple pieces and it takes the prediction of the first piece as the prediction for the word. The model is fine-tuned to minimize cross-entropy loss.

4.6. Experiments

We trained all classifiers described above using training data produced with XLM and FairSeq MT systems. Overall, classifiers trained with FairSeq translations performed very marginally better on out-of-domain data, with an average accuracy of 64.5%, compared to 64.3% for classifiers trained with XLM translations. In this section, we report only the results of classifiers trained with FairSeq translations, but both training sets produce very comparable results.

4.6.1. Monolingual task

Results on the monolingual task are reported in Table 4.2. Most accuracies are over the 50% that would be obtained by a random guess, albeit by a small margin on some conditions. Expectedly, the best performances are observed on in-domain data (TM), in which machine translations were produced by the same MT systems used to produce the classifiers' training data. Which of XLM or FairSeqwas used to produce test translations has little to no impact on performance, however. The highest accuracy (84.3%) is obtained on TM data by fine-tuning the FlauBERT pre-trained representations on the training material produced with XLM. Using this configuration, but classifying translations produced by DeepL only slightly reduces performance (82.4%). A similar trend is notable for the XLM-RoBERTa configuration (83% versus 81.9%), but any other approach — including other BERT-inspired solutions — leads to a notable decrease of accuracy otherwise.

HANS and EURO are the hardest test sets, where performances are often close to the random guess baseline. This suggests that translations produced by GT and DeepL on those datasets are very good and hard to distinguish from human translations. Part of this poor performance may be imputed to some extent to the mismatch between the system used to translate the classifiers's training material, and the one used for testing.

	TM			NEWS		CRAWL		HANS		EURO	
	X-	F-	D-	GT-	D-	GT-	D-	GT-	D-	GT-	D-
Feature-based classifiers:											
n-GRAM	76.0	76.6	81.4	66.6	72.6	59.2	61.9	47.2	49.6	53.6	56.3
KENLM	80.2	80.4	58.6	49.8	49.6	50	49.6	49.1	49.7	50.3	50.2
Deep-learning classifiers:											
bi-LSTM	64.5	62.7	53.3	60.8	59.3	57.7	55.7	**57.9**	55.5	58.5	57.4
LASER	55.9	56.3	58.4	54.8	54.5	54.5	53.9	54.7	50.5	54.1	53.6
Transformer-based classifiers:											
CamemBERT	83.7	**83.8**	73.8	68.9	**77.3**	**63.0**	**68.8**	52.3	**58.5**	56.6	60.5
XLM-RoBERTa	83.0	83.5	75.1	67.4	76.6	60.1	66.5	51.2	58.0	55.2	60.0
FlauBERT	**84.3**	82.2	**82.4**	**71.3**	77.0	64.8	66.4	51.7	53.8	**59.8**	**61.5**
XLM	79.9	77.5	72.3	69.8	73.2	60.3	61.0	49.9	50.1	54.9	56.0
mBERT	78.4	78.8	72.2	70.9	74.4	60.5	61.5	49.4	50.2	54.8	56.0

Tableau 4.2. Accuracy of classifiers on the monolingual classification task, on all test sets. X, F, D, and GT refer to the XLM, FairSeq, DeepL, and Google translation engines, respectively.

The lowest performances overall are recorded when classifying sentences produced by GT on the HANS dataset, where the best classifier only succeeds at a rate of 57.9%. Around 15% of automatic translations in this test set are identical to the reference one (see Table 4.4). Also, it is notorious that the GT system has been trained on Hansards, further complicating the task.

If we set apart those two test sets, we observe that BERT-like models provide better results than bi-LSTM and LASER ones. BERT models are systematically better at classifying DeepL translations than those produced by GT. We do not have a clear explanation for this.

The n-GRAM feature-based classifier is competitive with the LASER and bi-LSTM classifiers, but is slightly behind BERT-inspired classifiers. KENLM is clearly overfitting, delivering impressive results for such a simple device on in-domain data and systems, but failing to generalize to other settings.

The good performances we obtained on TM, when distinguishing translations produced by DeepL may be of interest to the language service provider that provided us with the data. It could for instance be used to diagnose translation providers that heavily rely on this system to produce their translations. The performance obtained on the NEWS and CRAWL test sets indicate that the automatic translations do have a signature that we can recognize to some extent, without even looking at the source sentence.

4.6.2. Bilingual Task

Table 4.3 shows accuracies obtained in the bilingual task, that is, when both the source sentence and the translation are considered. With a very few exceptions, all configurations benefit the extra input. For settings where the monolingual accuracies are high, the gains can be modest (for instance less than 2

	TM			NEWS		CRAWL		HANS		EURO	
	X-	F-	D-	GT-	D-	GT-	D-	GT-	D-	GT-	D-
Feature-based classifiers:											
n-GRAM	76.2	77.9	81.9	66.8	73.2	59.2	62.1	54.4	51.8	49.6	47.2
TMOP	62.7	62.9	59.8	63.4	62.9	61.1	57.2	54.8	57.8	51.2	50.1
Deep-learning classifiers:											
bi-LSTM	66.5	65.2	57.8	68.9	65.8	71.6	68.7	65.5	63.6	66.6	57.0
LASER	68.0	68.8	68.3	77.2	75.1	80.8	78.5	**73.5**	50.3	73.2	63.1
Transformer-based classifiers:											
CamemBERT	**87.5**	**87.6**	**84.6**	76.3	84.2	77.8	82.2	66.8	**73.1**	71.3	65.4
XLM-RoBERTa	86.7	85.8	81.2	76.2	82.5	77.5	79.7	67.2	68.5	69.8	63.3
FlauBERT	84.9	84.1	81.8	76.3	81.7	75.4	75.4	61.7	62.9	68.9	62.7
XLM	84.3	82.4	83.5	75.5	79.5	76.5	77.1	58.0	58.7	64.0	55.8
mBERT	86.6	83.9	82.9	**81.1**	**85.7**	**83.2**	**83.1**	70.6	58.3	**76.8**	**68.3**

Tableau 4.3. Accuracy of classifiers on the bilingual classification task, on all test sets.

points for FlauBERT on in-domain test sets), but otherwise, clear improvements are observable. For instance, on the HANS test sets, gains close to 20 points can be observed for some Transformer-based classifiers.

The more challenging datasets are now handled with an accuracy around 70% or above, while for the other test sets, the best performances are over 80%. Similarly to the monolingual task, Transformer-based classifiers are the best performers. The TMOP classifier overall underperforms the bi-LSTM and LASER ones. The n-GRAM classifier shows signs of overfitting, and delivers disappointing results on out-of-domain data.

4.7. Analysis

Table 4.4 shows the accuracy of the best performing classifiers for each test set, alongside the BLEU score of the XLM translation engine for that set. We anticipated that poor quality MT would be easier to detect, but BLEU score does not seem to correlate strongly with the classification performance. The bilingual task is unquestionably easier to tackle and for many test sets, including out-of-domain ones, the best classifier achieves an accuracy over 80%, a rather decent level of performance we did not anticipate at first, considering the relatively high quality of current NMT output.

Figure 4.1 shows the cumulative accuracy (y-axis) in the bilingual task calculated over the number of target sentences, sorted by the length of sentences (number of tokens). For all test sets and all classifiers, we observe that the longer the translation, the better the accuracy. This corroborates the findings of [Arase and Zhou, 2013], that longer sentences are easier to classify. This is likely explained by the fact that translations of short sentences are more likely to be similar to the human translation, and longer sentences likely contain more problems, further easing detection.

	=ref %	BLEU	Monolingual Task		Bilingual Task	
F-TM	6.1	39.3	83.8	(CAM)	87.6	(CAM)
X-TM	5.3	37.8	84.3	(FLAU)	87.5	(CAM)
GT-EURO	3.5	37.4	59.8	(FLAU)	76.8	(mBERT)
D-TM	4.8	36.2	82.4	(FLAU)	84.6	(CAM)
GT-HANS	15.5	34.9	57.9	(LSTM)	73.5	(LASER)
D-HANS	14.1	34.6	58.5	(CAM)	73.1	(CAM)
D-NEWS	1.8	33.4	77.3	(CAM)	85.7	(mBERT)
GT-NEWS	1.8	32.0	71.3	(FLAU)	81.1	(mBERT)
D-EURO	2.0	31.8	61.5	(FLAU)	68.3	(mBERT)
GT-CRAWL	1.5	25.2	63.0	(CAM)	83.2	(mBERT)
D-CRAWL	1.5	25.0	68.8	(CAM)	83.1	(mBERT)

Tableau 4.4. Accuracy of best classifier (in percentage) for each test set, in the monolingual and bilingual tasks, as a function of the (normalized) BLEU score. Classifier training data were produced with the FairSeq MT system. The best classifier is specified in parentheses next to its accuracy. Column “=ref %” indicates the percentage of sentences for which MT output is identical to the reference.

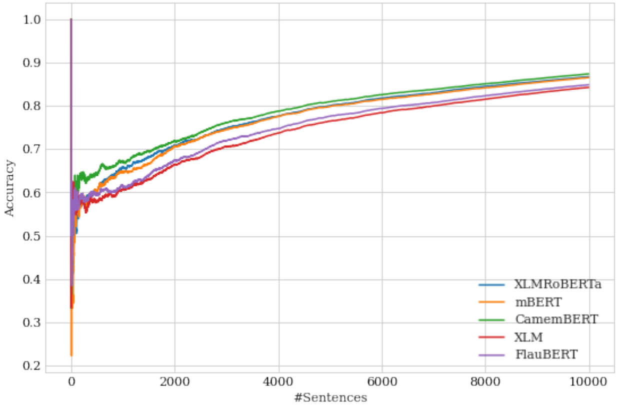


Figure 4.1. Cumulative accuracy in the bilingual task calculated over the number of target sentences produced by the XLM engine, sorted by their number of tokens.

We inspected the decisions made by our classifiers on some examples. We did notice machine translations involving problems with proper names and acronyms. We also occasionally found syntax problems in machine translations, such as example i) of Figure 4.2, which involves a failure in long-distance number agreement as well as a bad choice of pronoun. Also, we observed a strong tendency of machine translations to mimic the structure of the source sentence, as can be seen in most examples of Figure 4.2. This suggests that alignment features in the bilingual task could be useful. TMOP explicitly captures alignment information, but does not seem to make good use of it.

We were otherwise impressed by the overall quality of the MT, and rapidly realized how difficult it would be for human annotators to achieve a decent level of performance on this task. This is in line with the

i) **SRC** Are there any specific services being requested by SMEs that you are not able to provide for them or that you feel lie outside of your mandate?
HUM Les PME vous demandent-elles de leur fournir des services que vous ne pouvez leur donner ou qui, selon vous, échappent à votre mandat
NMT Y a-t-il des services particuliers demandés par les PME que vous ne pouvez pas leur fournir ou **que**, selon vous, ne **cadre** pas avec votre mandat ? (XLM, TM)

ii) **SRC** Until 2004, my parents met Nhan Thi Duong my ex-girlfriend and **asked for my daughter** Lan Thu Thi Le.
HUM Ils n'ont rencontré Nhan Thi Duong, mon ex-petite amie, qu'en 2004, et lui ont demandé **des nouvelles de ma fille**, Lan Thu Thi Le.
NMT Jusqu'en 2004, mes parents ont rencontré Nhan Thi Duong, mon ex-petite amie, et m'ont demandé **de me donner ma fille** Lan Thu Thi Le. (XLM, TM)

iii) **SRC** A bigger bloodbath seems inescapable if he does not **step down**.
HUM Il semble difficile d'échapper à un bain de sang plus important encore s'il n'accepte pas de démissionner.
NMT Un plus grand bain de sang semble inévitable s'il ne **se retire** pas. (DeepL, NEWS)

Figure 4.2. Examples of human and automatic translations. Underlined passages identify problems and bold ones their corresponding parts.

observations of Arase and Zhou [2013], who report lower performances for humans than for machines at detecting translations produced by statistical phrase-based MT.

To better understand the type of information our classifiers base their decisions on, we inspected cases where the human translation is predominantly classified as such by our classifiers,[23] and the machine translation counterpart is predominantly recognized as a machine translation. Then, we manually produced minimal pairs, that is, as small as possible variants of the automatic translation, to see at which point the classifiers were changing their decisions from machine to human, thus allowing us to see which signals they react to.

We found that in most cases, modifying only a few words (often only one) of the automatic translation is enough for the classifier to reverse its decision. Some cases involved normalizations that our post-processing script (see Section 4.4.3) fails to take into account. Among those, we noted the presence of a hyphen symbol produced by DeepL on the NEWS data set, different from the one used in human translations. We also noted a few cases involving typographical preferences.

For instance, on the EURO test set, removing a space in section numbering produced by XLM (*e.g.* "`5 c)`" versus "`5c)`") sometimes suffices to make our classifiers believe the translation is human. Also, removing a capital letter (or sometimes adding one) may reverse the classifier's decision.

[23] By "predominant", we mean that at least 15 out of our 18 classifiers agreed.

Of course, such normalization issues are in a way deceptive since although they do help decision making, they do not have much to do with translation quality. In any case, the most frequent situation involves lexical choices. Sometimes, it is easy to blame the translation engine, as in example ii), but sometimes it is less, as in example iii).

4.8. Conclusion

In this study, we implemented 18 classifiers to detect machine-translated texts, and evaluated their performance on several test sets, containing translations produced by different state-of-the-art NMT systems. Overall, we found that classifiers with access to both the source sentence and the translation perform better than those with access to the translation alone. Our classifiers achieve accuracies above 80% on several test sets and always surpass a random baseline. Our analysis reveals that, despite of our efforts to normalize translations, artifacts still exist in the data that could explain in part our relatively high classifier accuracies. But in general, it appears that NMT systems do elicit signatures that can be recognized by automatic methods. Often, a single lexical choice gives away the automatic nature of the translation, even when the translation looks fluent from a language model point of view.

In future work, we hope to produce better MT detectors by creating training data using a wider variety of MT systems. Another question we would like to examine is to what extent it is possible to detect post-edited translations, i.e. machine translations manually edited by human translators.

Chapter 5

OSTI: An Open-Source Translation-memory Instrument

Contribution

David Alfonso-Hermelo [1], Philippe Langlais[1], Shivendra Bhardwaj[1], Michel Simard [2], Cyril Goutte[2] and Gabriel Bernier-Colborne[2]. OSTI: An Open-Source Translation-memory Instrument, 2020. Submitted at 28th International Conference on Computational Linguistics (COLING'2020).
{shivendra.bhardwaj, david.alfonso.hermelo, philippe.iro}@umontreal.ca
{Michel.Simard, Cyril.Goutte, Gabriel.Bernier-Colborne}@nrc-cnrc.gc.ca

In this article, David did the integration work and wrote the paper with Professor Philippe Langlais. I assisted David with the integration of the deep learning tool. The NRC-CNRC team helped in proofreading.

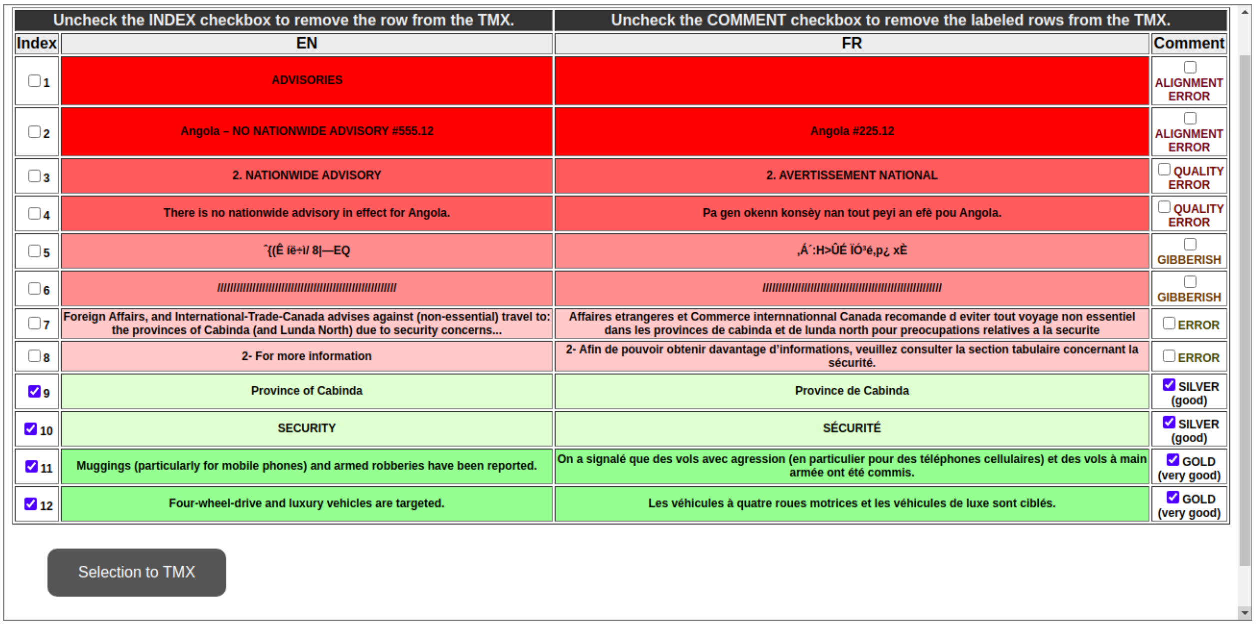
Abstract

We present OSTI: a free open-source tool to process and visualize a pair of bilingual documents (original and translation) into automatically labeled sentence pairs. This can be used by translation professionals as a human-accessible quality evaluation tool, as a pre-processing step for human annotation as well as an intermediate step to populate a Translation Memory.

5.1. Introduction

The development of Computer-Assisted Translation (CAT) tools started gaining popularity in the mid 1980s. Since then, the elaboration of new and more sophisticated CAT tools has not ceased to increase. Given that training data is essential to the development of high-end automatic models, there has been great academic and (sometimes) corporate efforts to make large and clean Translation Memories (TM) and translation data sets (bilingual and monolingual dictionaries, terminologies, etc.) publicly available. Works such as [Koehn, 2005, Tiedemann, 2012, Steinberger et al., 2013] have greatly facilitated the elaboration of pioneering CAT tools *freely* available today.

[1] Researchers from RALI
[2] Researchers from NRC-CNRC



Uncheck the INDEX checkbox to remove the row from the TMX.		Uncheck the COMMENT checkbox to remove the labeled rows from the TMX.	
Index	EN	FR	Comment
□ 1	ADVISORIES		□ ALIGNMENT ERROR
□ 2	Angola – NO NATIONWIDE ADVISORY #555.12	Angola #225.12	□ ALIGNMENT ERROR
□ 3	2. NATIONWIDE ADVISORY	2. AVERTISSEMENT NATIONAL	□ QUALITY ERROR
□ 4	There is no nationwide advisory in effect for Angola.	Pa gen okenn konsèy nan tout peyi an efè pou Angola.	□ QUALITY ERROR
□ 5	ˆ{(Ê íë÷i/ 8|—EQ	,Á´:H>ÛÉ ÏÓ³é,p¿ xÈ	□ GIBBERISH
□ 6	//	//	□ GIBBERISH
□ 7	Foreign Affairs, and International-Trade-Canada advises against (non-essential) travel to: the provinces of Cabinda (and Lunda North) due to security concerns...	Affaires etrangeres et Commerce internnationnal Canada recomande d eviter tout voyage non essentiel dans les provinces de cabinda et de lunda north pour preocupations relatives a la securite	□ ERROR
□ 8	2- For more information	2- Afin de pouvoir obtenir davantage d'informations, veuillez consulter la section tabulaire concernant la sécurité.	□ ERROR
☑ 9	Province of Cabinda	Province de Cabinda	☑ SILVER (good)
☑ 10	SECURITY	SÉCURITÉ	☑ SILVER (good)
☑ 11	Muggings (particularly for mobile phones) and armed robberies have been reported.	On a signalé que des vols avec agression (en particulier pour des téléphones cellulaires) et des vols à main armée ont été commis.	☑ GOLD (very good)
☑ 12	Four-wheel-drive and luxury vehicles are targeted.	Les véhicules à quatre roues motrices et les véhicules de luxe sont ciblés.	☑ GOLD (very good)

Figure 5.1. Screenshot of OSTI's HTML visualization. Each sentence pair is presented in different colours according to their estimated quality (see text for details). As a default, `Gold` and `Silver` (right column) sentence pairs are selected for conversion to TMX, but it is up to the user to change this selection by removing/adding individual sentence pairs (*Index* checkbox on the left) or by removing/adding all sentence pairs having the same label (*Comment* checkbox on the right).

Nevertheless, a problem arises when users have their own proprietary data and want to use it instead of the general and publicly available one. Multiple companies offer to tailor-made an exclusive TM using proprietary data but these services can be black-boxes with untraceable quality, unaffordable to the more humble translators or translation companies, or apprehensive to companies working with sensitive data. Having this in mind, we designed a simple yet (hopefully) useful open-source tool which takes as input a pair of supposedly parallel documents in English and French.[3] These are segmented, aligned into units whose quality is automatically inspected, labeled, and presented as an easy to consume HTML (an example of which is reported in Figure 5.1). They can subsequently be distilled into a TMX format.

Our tool can be used as a human-accessible quality evaluation tool, as a pre-processing step for human annotation, as well as an intermediate step to populate a Translation Memory.

5.2. System overview

We conceived OSTI in a modular fashion, trying to make the integration of new components, tools or language pairs as easy as possible. Its has been currently tested on the French-English language pair (our use case), but it should apply with minor or no adaptation to other language pairs.

The overall pipeline, depicted in Figure 5.2, contains 4 modules: a) we first segment each text into sentences, that b) we align at the sentence level; c) the sentence pairs (SPs) are then classified into good

[3] We targeted English and French languages, but arguably many components could be used for other language pairs. To keep it simple, we also assume that documents are converted into text prior to using OSTI.

or bad SPs; d) the SPs are further labelled into 6 classes for the sake of human readability on an HTML interface that allows to export the automatically or manually selected SPs into a TMX file.

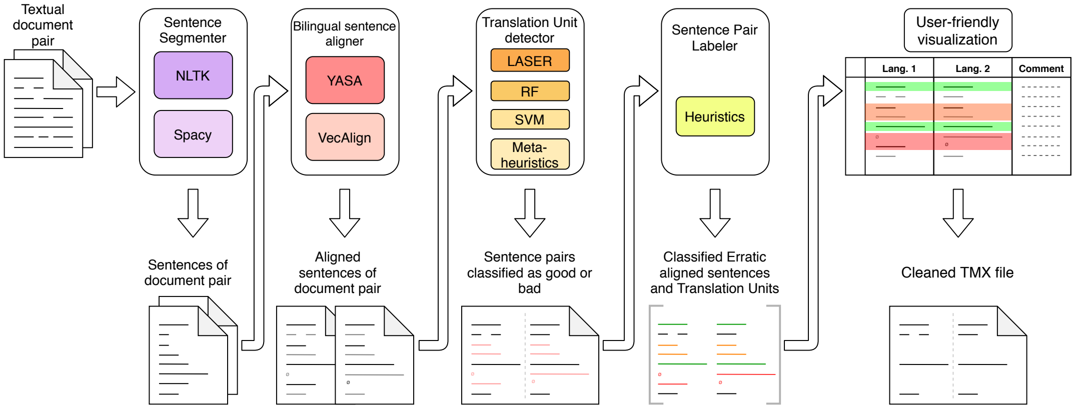

Textual document pair
Sentence Segmenter
NLTK
Spacy
Bilingual sentence aligner
YASA
VecAlign
Translation Unit detector
LASER
RF
SVM
Meta-heuristics
Sentence Pair Labeler
Heuristics
User-friendly visualization
Lang. 1
Lang. 2
Comment
Sentences of document pair
Aligned sentences of document pair
Sentence pairs classified as good or bad
Classified Erratic aligned sentences and Translation Units
Cleaned TMX file

Figure 5.2. OSTI pipeline, from document pair to visualization. Top (darker) components are defaults.

5.2.1. Sentence Segmenter

Even though the task of sentence segmentation is not a very enticing one, it is crucial to have a clean sentence segmentation to start with in order to reduce the subsequent tasks. We use the NLTK sentence tokenizer as a default, although we also considered Spacy [spaCy, 2017] and the `Mediacloud` Sentence Splitter.[4] A small empirical analysis showed that NLTK outputs the best out-of-the-box results for our specific language pair, we therefore selected it as our default segmenter.

5.2.2. Bilingual Sentence Aligner

We benchmarked two very different tools that have both shown to be accurate and robust: YASA [Lamraoui and Langlais, 2013] and VECALIGN [Thompson and Koehn, 2019]. The former system is very similar to `HunAlign` [Varga et al., 2007], that is, a sentence-length score [Gale and Church, 1993a] enhanced with a cognate-based one [Simard et al., 1992], and has been reported faster and more accurate than more elaborated systems such as BMA [Moore, 2002]. The VECALIGN system uses a more innovative scoring function specially made to work with sentence embedding vectors. Although this system is said to accommodate embeddings from various toolkits, we used the recommended multilingual Language-Agnostic SEntence Representations (LASER) [Artetxe and Schwenk, 2019b].

Since our target language pair is French-English, we compared both aligners using the BAF corpus benchmark [Simard, 1998] which contains 11 document pairs of 4 different genres (Literary, Institutional, Scientific, and Technical) segmented into approximately 25k sentences (in each language) aligned and manually checked. To evaluate the aligners, we used the benchmark's metrics of precision, recall and F1 at the alignment pair level and at the sentence level. See Langlais et al. [1998] for a description of those metrics.

[4] https://github.com/berkmancenter/mediacloud-sentence-splitter (based on the implementation by Koehn [2005])

		Literacy		Institutional		Scientific		Technical		**All doc.**	
		YASA	VECA	YASA	VECA	YASA	VECA	YASA	VECA	YASA	VECA
align.	Prec.	0.59	0.61	0.94	0.95	0.86	0.86	0.85	0.86	0.86	**0.87**
	Rec.	0.74	0.71	0.95	0.95	0.93	0.91	0.96	0.95	**0.92**	0.91
	F_1	0.65	0.65	0.94	0.95	0.89	0.88	0.90	0.90	**0.89**	**0.89**
sent.	Prec.	0.88	0.87	0.98	0.98	0.99	0.98	0.99	0.98	**0.98**	0.97
	Rec.	0.79	0.84	0.94	0.95	0.86	0.86	0.06	0.06	0.81	**0.82**
	F_1	0.83	0.86	0.96	0.96	0.92	0.91	0.11	0.11	**0.85**	**0.85**

Tableau 5.1. Alignment and sentence Precision, Recall and F_1 scores of YASA and VECALIGN as a function of text genre on the BAF benchmark. The last columns shows the average over all document pairs, over all genres.

Results are reported in Table 5.1. Two observations can be made. First, both systems deliver very similar performances. Second, the performance varies substantially depending on the text genre, the worst setting is when aligning literary texts.[5] We refer the reader to Xu et al. [2015] for extensive comparisons of alignment techniques on literary texts. Because both systems perform on par, we selected YASA as our default sentence aligner since is it much lighter than VECALIGN in terms of hardware (CPU versus GPU).[6]

5.2.3. Translation Unit Detector

This component decides whether a given sentence pair contains a problem or not. Sentence pairs that are classified as good are promoted to *Translation Units* (TU) and can be saved in TMX format for further consumption. In order to do so, we implemented 5 families of classifiers, described in details elsewhere Anonymous [2020]: a) a heuristic-based approach (META-HEURISTICS in Figure 5.2) involving 13 heuristics also described in Sub-section 5.2.4; b) statistical classifiers using 62 features (SVM and RF in Fig. 5.2) and c) a cleaning method devised in-house on top of the LASER model developed and pre-trained by Artetxe and Schwenk [2019b].

We compared those classifiers in terms of accuracy on a manually annotated proprietary corpus of 2021 sentence pairs and found LASER to largely outperform the feature-based classifiers (84% accuracy vs. 63%) and meta-heuristics (42%). Considering that LASER is unsupervised, and relatively fast to use, we selected it as our default detector.

5.2.4. Sentence Pair Labeler

There are many reasons why sentence pairs can be detected erroneous, including bad sentence alignment (often caused by sentence segmentation issues), errors in translations (calques, false friends, etc.), as well as encoding issues (which happen in complex organisations due to numerous format manipulations). All of these are considered errors but not all are equally important to the translation professional. Therefore, we take an extra step into further labelling sentences pairs into 6 labels described below. We do this by taking

[5] There is one pair of texts in BAF belonging to this category which is a novel of Jules Vernes where the English version is abridged, which confuses the dynamic programming optimization driving each approach.

[6] VECALIGN accommodates CPU computations, at the expense of much slower response time.

advantage of the 13 heuristics used in the META-HEURISTIC component of the TU detector. Each heuristic detects a specific type of error.

Aligt: qualifies a major mismatch, such as numerical entity issues (row 2 in Figure 5.1), sentence length (row 1 in Figure 5.1), etc. In total, 8 heuristics are used.

Quality: characterizes mistakes identified by 5 heuristics: misspelling issues (row 4 in Figure 5.1), false-friends, sentences that are part of table of content or indexes and are often misaligned (row 3 in Figure 5.1), and non-translated target sentences (which happens when documents are partly translated).

Gibberish: qualifies sentence pairs that contain mainly gibberish (row 6 in Figure 5.1), sometimes due to encoding issues (row 5 in Figure 5.1).

Error: is used when an alignment problem is detected but can not be attributed to a specific cause. Row 7 (punctuation mismatch and misspelling issues) and row 8 (table of content detection and length mismatch) in Figure 5.1 are examples, where both Aligt and Quality compete.

Silver: is used when the TU detector classifies a sentence pair as good but at least one heuristic indicates the presence of a problem (row 9 and 10 in Figure 5.1).

Gold: qualifies SPs that are classified good by the detector, and for which no heuristic indicates any problem (row 11 and 12 in Figure 5.1).

5.2.5. Technical details

Our Github project[7] requires the installation of a small amount of modules (such as NLTK, NUMPY, FAISS or PYTORCH) that are specified in our requirements file and are easily installed using the standard pip package manager or by executing the setup file. It also requires the installation of more complex tool-kits (i.e., VECALIGN and LASER). The YASA tool is provided in compiled form and has only been tested to work on (multiple) Debian-based Linux distributions. Finally, it also includes all the in-house algorithms/implementations written in Python as well as some pre-trained models mentioned in Section 5.2.3. This allows to run OSTI without having installed LASER, only using a functional yet lesser classifier.

To ensure its viability, we fully tested OSTI on a computer with 30Gb of RAM, a 12-core CPU and a GeForce GT 1030 GPU (used by LASER and VECALIGN) using a Debian-based Linux Operating System.

We also measured its response time of on 2 document pairs containing around 900 sentences each (1 800 sentences/160k characters in total). On average, sentence segmentation took less than 0.5 seconds. Sentence alignment took 13 seconds with YASA (on CPU) and 17 seconds with VECALIGN (with GPU). For the TU detector, the fastest is the heuristic-based approach (112 seconds, CPU), followed by LASER (117 seconds, GPU) and the 2 feature based classifiers (SVM: 131 seconds, RF: 239 seconds, both using a single CPU). Finally, to save time, the sentence pair labeler is run concurrently to the TU detector.

5.3. Conclusion

We presented OSTI, an open-source tool which detects good from bad sentence pairs in a French-English pair of (supposedly) parallel documents. These are then further labelled into a set of 6 labels that can be inspected with a simple Web browser and easily transformed into a TMX file. OSTI can be used as

[7] https://github.com/dahrs/OSTI

a human-accessible quality evaluation tool, as a pre-processing step for human annotation, as well as an intermediate step to populate a Translation Memory. We are aware of proprietary solutions, but there is a striking absence of open-source and peer reviewed such systems.

Currently, we targeted the English-French langage pair, which we plan to revisit. We do not anticipate much difficulties since most components involved in OSTI are arguably language agnostic. Also, we distribute a simple batch pipeline, while for professional use, a client-server application may be more appropriate. In benchmarking embedded components, we were surprised by the fact that a simple sentence aligner was performing on par with a more recent one relying on sentence embeddings. We plan to revisit this benchmarking on other language pairs and conditions.

Chapter 6

Conclusion

This thesis tried to address a broad section of noise present in a professional Translation Memory (TM). We have presented an extensive comparison of techniques such as heuristics, feature-based and deep-learning-based, which were applied to two novel problems: First, cleaning a professional TM of good quality and the second was detecting Machine Translation (MT) output, discussed in Chapter 3 and 4 respectively. All the experiments were conducted at the sentence level of a parallel corpus with LASER and plain old technologies. The outcome suggests that the LASER is as good, and much faster, as discussed in Chapter 5.

We have developed convincing tools for the two problems, as well as an interface for professional translators at the Translation Bureau of Canada. The three articles are submitted to a very selective conference (acceptance rate around 20%). As of now, we are unsure if the articles will get accepted, but both the papers are legitimate and demonstrate interesting and informative experiments that were not available before.

6.1. Thesis Contribution

6.1.1. Noise Cleaning

We showcase that we can detect and remove noise from an almost clean TM (generated by professional translators) using both supervised and unsupervised learning methods. We demonstrate a gain of +1.27 and +0.98 BLEU scores from supervised and unsupervised learning methods respectively. We also report a significant gain of +1.08 BLEU score over a State-Of-The-Art (SOTA), off-the-shelf TM cleaning system using an intersection of supervised and unsupervised learning methods (∩ALL from 3.3). During the analysis, we note that, apart from the self-evident noise like sentence misalignment, the model was generalized enough to capture noise like URL mismatch and gibberish words.

Our proposed tool named "OSTI: An Open-Source Translation-memory Instrument" (Chapter 5) is an open-source web-based tool that can be used as a pre-processing step for human annotation. It annotates not only the problematic sentence pairs but also helps users to drill down further into six critically identified classes of problem.

6.1.2. MT Detection

To the best of our knowledge, this is the first time extensive experiments are reported on distinguishing human from machine translation produced by a SOTA Neural Machine Translation (NMT) systems (in-domain and out-of-domain) under both monolingual and bilingual settings. Also, this research is one of its kind to deploy transformers-based language models for this task. We achieve an accuracy of 80% on several test sets (in-domain and out-of-domain), which is way above our random baseline.

Although the NMT systems are capable enough to produce reliable translations, it is still difficult for the NMT systems to capture the entire coherence and context of the document before translating the texts. To appreciate the MT detection results, we have manually analyzed a small sample, which shows that the models (transformers-based) were also looking at the structure and the lexical choice before making the decisions (from Fig 4.2).

6.2. Future Improvement

6.2.1. Noise Cleaning

We want to examine more subtle and granular noise (manually look into the data) and design sophisticated heuristics. This will help us to collect such noise in the initial stage which will enrich the training matarial. Also, an in-depth exploration is needed in the direction of artificially designing a more refined negative sample than the one we introduce in the first paper (Chapter 3). It will enhance the model to point out the translations produced with the wrong lexical choice.

We know that LASER produces false positives (based on manual annotation by a language expert) discussed in Section 3.6.3.5, and that the LASER base model is not better than YASA (Chapter 5), which is why more work is required to boost the LASER solution. As BERT flavoured language models worked well in the MT detection task, a follow-up study could be made to investigate its performance for the noise detection task.

6.2.2. MT Detection

Our work pointed out the brilliantness of the decision taken by models (discussed in Section 4.7), but greater in-depth analysis needs to be conducted. Also, it remains to see if machine-translated texts are prevalent in the MDMT corpus (TM provided by the Translation Bureau of Canada), or whether the presence of such texts are occasional. Additionally, a follow-up can be made to examine whether the models can detect post-edited translations, i.e. machine translations manually edited by human translators.

To produce a more generalized MT detection model, one can collect more data from numerous NMT models to train the classifiers. Also, including publicly available data (such as EURO and HANS) to train the classifier can be examined. We conducted our experiments at the sentence level; a follow-up problem is one using context, i.e. at the paragraph level.

Bibliography

National research council canada. https://nrc.canada.ca/en/research-development/products-services/technical-advisory-services/multilingual-text-processing.

R. Aharoni, M. Koppel, and Y. Goldberg. Automatic detection of machine translated text and translation quality estimation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 289–295, Baltimore, Maryland, June 2014. Association for Computational Linguistics. doi: 10.3115/v1/P14-2048. URL https://www.aclweb.org/anthology/P14-2048.

S. Ananthakrishnan, R. Prasad, and P. Natarajan. Phrase alignment confidence for statistical machine translation. In *INTERSPEECH*, 2010.

Anonymous. Cleaning a(n almost) clean institutional translation memory. submitted, 2020.

A. Antonova and A. Misyurev. Building a web-based parallel corpus and filtering out machine-translated text. In *Proceedings of the 4th Workshop on Building and Using Comparable Corpora: Comparable Corpora and the Web*, pages 136–144, Portland, Oregon, June 2011. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W11-1218.

Y. Arase and M. Zhou. Machine translation detection from monolingual web-text. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1597–1607, Sofia, Bulgaria, Aug. 2013. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P13-1157.

M. Artetxe and H. Schwenk. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610, 2019a. URL https://transacl.org/ojs/index.php/tacl/article/view/1742.

M. Artetxe and H. Schwenk. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610, 2019b.

D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate, 2014. URL http://arxiv.org/abs/1409.0473. cite arxiv:1409.0473Comment: Accepted at ICLR 2015 as oral presentation.

L. Bahl, F. Jelinek, and R. Mercer. A maximum likelihood approach to continuous speech recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-5:179 – 190, 04 1983. doi: 10.1109/TPAMI.1983.4767370.

J. K. Baker. Stochastic modeling for automatic speech understanding. *R. A. Reddy (ed.), Speech Recognition. New York: Academic Press.*, 1979.

M. Baker, G. Francis, and E. Tognini-Bonelli. *'Corpus Linguistics and Translation Studies: Implications and Applications'*. John Benjamins Publishing Company, Netherlands, 1993.

M. Bansal, C. Quirk, and R. Moore. Gappy phrasal alignment by agreement. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1308–1317, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/P11-1131`.

E. Barbu. Spotting false translation segments in translation memories. In *Proceedings of the Workshop on Natural Language Processing for Translation Memories*, pages 9–16, 2015.

E. Barbu. Ensembles of classifiers for cleaning web parallel corpora and translation memories. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 71–77, Varna, Bulgaria, Sept. 2017. INCOMA Ltd. doi: 10.26615/978-954-452-049-6_011. URL `https://doi.org/10.26615/978-954-452-049-6_011`.

E. Barbu, C. Parra Escartín, L. Bentivogli, M. Negri, M. Turchi, C. Orasan, and M. Federico. The first automatic translation memory cleaning shared task. *Machine Translation*, 30(3–4):145–166, Dec. 2016. ISSN 0922-6567. doi: 10.1007/s10590-016-9183-x. URL `https://doi.org/10.1007/s10590-016-9183-x`.

M. Baroni and S. Bernardini. A new approach to the study of translationese: Machine-learning the difference between original and translated text. *Lit. Linguistic Comput.*, 21:259–274, 2006.

L. E. Baum. An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes. In O. Shisha, editor, *Inequalities III: Proceedings of the Third Symposium on Inequalities*, pages 1–8, University of California, Los Angeles, 1972. Academic Press.

V. Becher. When and why do translators add connectives?: A corpus-based study. *Target*, 23:26–47, 01 2011. doi: 10.1075/target.23.1.02bec.

G. Bernier-Colborne and C.-k. Lo. NRC parallel corpus filtering system for WMT 2019. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 252–260, Florence, Italy, Aug. 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-5434. URL `https://www.aclweb.org/anthology/W19-5434`.

S. Blum-Kulka and E. A. Levenston. Universals of lexical simplification. strategies in interlanguage communication. page 119 – 139, 10 1983.

P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001. ISSN 0885-6125. doi: 10.1023/A:1010933404324. URL `http://dx.doi.org/10.1023/A%3A1010933404324`.

P. F. Brown, J. Cocke, S. A. Della Pietra, V. J. Della Pietra, F. Jelinek, J. D. Lafferty, R. L. Mercer, and P. S. Roossin. A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85, 1990. URL `https://www.aclweb.org/anthology/J90-2002`.

P. F. Brown, J. C. Lai, and R. L. Mercer. Aligning sentences in parallel corpora. In *Proceedings of the 29th Annual Meeting on Association for Computational Linguistics*, ACL '91, page 169–176, USA, 1991. Association for Computational Linguistics. doi: 10.3115/981344.981366. URL `https://doi.org/10.3115/981344.981366`.

P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, and R. L. Mercer. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311, 1993a. URL https://www.aclweb.org/anthology/J93-2003.

P. F. Brown, V. J. D. Pietra, S. A. D. Pietra, and R. L. Mercer. The mathematics of statistical machine translation: Parameter estimation. *Comput. Linguist.*, 19(2):263–311, June 1993b. ISSN 0891-2017.

C. Buck and P. Koehn. Uedin participation in the 1st translation memory cleaning shared task. In *Proceedings of the 2nd Workshop on Natural Language Processing for Translation Memories (NLP4TM)*, 2016.

K. Bundgaard and T. Christensen. Is the concordance feature the new black? a workplace study of translators' interaction with translation resources while post-editing tm and mt matches. *Journal of Specialised Translation*, 31(31):14–37, Jan. 2019. ISSN 1740-357X.

D. Carter and D. Inkpen. Searching for poor quality machine translated text: Learning the difference between human writing and machine translations. In L. Kosseim and D. Inkpen, editors, *Advances in Artificial Intelligence*, pages 49–60, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. ISBN 978-3-642-30353-1.

W. B. Cavnar and J. M. Trenkle. N-gram-based text categorization. In *Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval*, pages 161–175, 1994.

V. Chaudhary, Y. Tang, F. Guzmán, H. Schwenk, and P. Koehn. Low-resource corpus filtering using multilingual sentence embeddings. *arXiv preprint arXiv:1906.08885*, 2019.

B. Chen, R. Kuhn, G. Foster, C. Cherry, and F. Huang. Bilingual methods for adaptive training data selection for machine translation. 2016.

A. Chesterman. Beyond the particular. in: Mauranen, a., kujamaki, p. (eds) translation universals: Do they exist? amsterdam: Benjamins. page 33 – 49, 2004.

K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *CoRR*, abs/1409.1259, 2014. URL http://arxiv.org/abs/1409.1259.

F. Chollet et al. Keras. https://keras.io, 2015.

C. Chu, T. Nakazawa, and S. Kurohashi. Constructing a Chinese—Japanese parallel corpus from Wikipedia. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 642–647, Reykjavik, Iceland, May 2014. European Language Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2014/pdf/21_Paper.pdf.

C. Chu, R. Dabre, and S. Kurohashi. Parallel sentence extraction from comparable corpora with neural network features. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 2931–2935, Portorož, Slovenia, May 2016. European Language Resources Association (ELRA). URL https://www.aclweb.org/anthology/L16-1468.

A. Conneau and G. Lample. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems*, pages 7057–7067, 2019a.

A. Conneau and G. Lample. Cross-lingual language model pretraining. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 7059–7069. Curran Associates, Inc., 2019b. URL http://papers.nips.cc/paper/8928-cross-lingual-language-model-pretraining.pdf.

L. Cui, D. Zhang, S. Liu, M. Li, and M. Zhou. Bilingual data cleaning for SMT using graph-based random walk. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 340–345, Sofia, Bulgaria, Aug. 2013. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/P13-2061`.

Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier. Language modeling with gated convolutional networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 933–941. JMLR.org, 2017.

J. G. de Souza, M. Turchi, and M. Negri. Machine translation quality estimation across domains. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 409–420, 2014.

J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018a. URL `http://arxiv.org/abs/1810.04805`.

J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018b. URL `http://arxiv.org/abs/1810.04805`.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL `https://www.aclweb.org/anthology/N19-1423`.

C. Dyer, V. Chahuneau, and N. A. Smith. A simple, fast, and effective reparameterization of IBM model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648, Atlanta, Georgia, June 2013. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/N13-1073`.

S. Edunov, M. Ott, M. Auli, and D. Grangier. Understanding back-translation at scale. *CoRR*, abs/1808.09381, 2018. URL `http://arxiv.org/abs/1808.09381`.

C. Espana-Bonet, A. C. Varga, A. Barron-Cedeno, and J. van Genabith. An empirical analysis of nmt-derived interlingual embeddings and their use in parallel sentence identification. *IEEE Journal of Selected Topics in Signal Processing*, 11(8):1340–1350, Dec 2017. ISSN 1941-0484. doi: 10.1109/jstsp.2017.2764273. URL `http://dx.doi.org/10.1109/JSTSP.2017.2764273`.

P. Fung and P. Cheung. Mining very-non-parallel corpora: Parallel sentence and lexicon extraction via bootstrapping and e. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 57–63, Barcelona, Spain, July 2004a. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/W04-3208`.

P. Fung and P. Cheung. Multi-level bootstrapping for extracting parallel sentences from a quasi-comparable corpus. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 1051–1057, Geneva, Switzerland, aug 23–aug 27 2004b. COLING. URL `https://www.aclweb.org/anthology/C04-1151`.

P. Fung and L. Y. Yee. An IR approach for translating new words from nonparallel, comparable texts. In *COLING 1998 Volume 1: The 17th International Conference on Computational Linguistics*, 1998. URL https://www.aclweb.org/anthology/C98-1066.

W. A. Gale and K. W. Church. A program for aligning sentences in bilingual corpora. *Comput. Linguist.*, 19(1):75–102, Mar. 1993a.

W. A. Gale and K. W. Church. A program for aligning sentences in bilingual corpora. *Computational linguistics*, 19(1):75–102, 1993b.

Q. Gao and S. Vogel. Parallel implementations of word alignment tool. In *Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, pages 49–57, Columbus, Ohio, June 2008a. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W08-0509.

Q. Gao and S. Vogel. Parallel implementations of word alignment tool. In *Software engineering, testing, and quality assurance for natural language processing*, pages 49–57, 2008b.

R. Garside, G. Leech, and G. Sampson. The computational analysis of english: A corpus-based approach. *Language*, 65, 06 1989. doi: 10.2307/415358.

J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 1243–1252. JMLR.org, 2017.

C. Goutte, M. Carpuat, and G. Foster. The impact of sentence alignment errors on phrase-based machine translation performance. In *Proc. of AMTA*, 2012.

F. Grégoire and P. Langlais. Extracting parallel sentences with bidirectional recurrent neural networks to improve machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1442–1453, Santa Fe, New Mexico, USA, Aug. 2018. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/C18-1122.

J. Grover and P. Mitra. Bilingual word embeddings with bucketed CNN for parallel sentence extraction. In *Proceedings of ACL 2017, Student Research Workshop*, pages 11–16, Vancouver, Canada, July 2017. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P17-3003.

K. Heafield. Kenlm: Faster and smaller language model queries. In *In Proc. of the Sixth Workshop on Statistical Machine Translation*, 2011.

K. Heafield, I. Pouzyrevsky, J. H. Clark, and P. Koehn. Scalable modified Kneser-Ney language model estimation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 690–696, Sofia, Bulgaria, Aug. 2013. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P13-2121.

M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf. Support vector machines. *IEEE Intelligent Systems and their Applications*, 13(4):18–28, 1998.

S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, Nov. 1997a. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. URL http://dx.doi.org/10.1162/neco.1997.9.8.1735.

S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997b.

I. Ilisei, D. Inkpen, G. Corpas Pastor, and R. Mitkov. Identification of translationese: A machine learning approach. In *Proceedings of the 11th International Conference on Computational Linguistics and Intelligent Text Processing*, CICLing'10, page 503–511, Berlin, Heidelberg, 2010. Springer-Verlag.

ISBN 3642121152. doi: 10.1007/978-3-642-12116-6_43. URL https://doi.org/10.1007/978-3-642-12116-6_43.

K. Imamura and E. Sumita. Bilingual corpus cleaning focusing on translation literality. In *INTERSPEECH*, 2002.

K. Imamura, E. Sumita, and Y. Matsumoto. Feedback cleaning of machine translation rules using automatic evaluation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, ACL '03, page 447–454, USA, 2003. Association for Computational Linguistics. doi: 10.3115/1075096.1075153. URL https://doi.org/10.3115/1075096.1075153.

M. Jalili Sabet, M. Negri, M. Turchi, and E. Barbu. An unsupervised method for automatic translation memory cleaning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 287–292, Berlin, Germany, Aug. 2016a. Association for Computational Linguistics. doi: 10.18653/v1/P16-2047. URL https://www.aclweb.org/anthology/P16-2047.

M. Jalili Sabet, M. Negri, M. Turchi, J. G. C. de Souza, and M. Federico. TMop: a tool for unsupervised translation memory cleaning. In *Proceedings of ACL-2016 System Demonstrations*, pages 49–54, Berlin, Germany, Aug. 2016b. Association for Computational Linguistics. doi: 10.18653/v1/P16-4009. URL https://www.aclweb.org/anthology/P16-4009.

M. Joshi, D. Chen, Y. Liu, D. S. Weld, L. Zettlemoyer, and O. Levy. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8: 64–77, 2020. doi: 10.1162/tacl_a_00300. URL https://doi.org/10.1162/tacl_a_00300.

M. Junczys-Dowmunt. Dual conditional cross-entropy filtering of noisy parallel corpora. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 888–895, Belgium, Brussels, Oct. 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-6478. URL https://www.aclweb.org/anthology/W18-6478.

M. Juuti, B. Sun, T. Mori, and N. Asokan. Stay on-topic: Generating context-specific fake restaurant reviews, 2018.

N. Kalchbrenner and P. Blunsom. Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1700–1709, Seattle, Washington, USA, Oct. 2013. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D13-1176.

S. Khadivi and H. Ney. Automatic filtering of bilingual corpora for statistical machine translation. In A. Montoyo, R. Muńoz, and E. Métais, editors, *Natural Language Processing and Information Systems*, pages 263–274, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

H. Khayrallah and P. Koehn. On the impact of various types of noise on neural machine translation. *arXiv preprint arXiv:1805.12282*, 2018.

H. Khayrallah, H. Xu, and P. Koehn. The JHU parallel corpus filtering systems for WMT 2018. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 896–899, Belgium, Brussels, Oct. 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-6479. URL https://www.aclweb.org/anthology/W18-6479.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In Y. Bengio and Y. LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9,* *2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980.

P. Koehn. Europarl: A parallel corpus for statistical machine translation. In *MT Summit X*, pages 79–86, 2005.

P. Koehn, F. J. Och, and D. Marcu. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, page 48–54, USA, 2003. Association for Computational Linguistics. doi: 10.3115/1073445.1073462. URL https://doi.org/10.3115/1073445.1073462.

P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P07-2045.

M. Koppel and N. Ordan. Translationese and its dialects. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1318–1326, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P11-1132.

T. Kudo and J. Richardson. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium, Nov. 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-2012. URL https://www.aclweb.org/anthology/D18-2012.

D. Kurokawa, C. Goutte, and P. Isabelle. Automatic detection of translated text and its impact on machine translation. 2009.

G. Lample and A. Conneau. Cross-lingual language model pretraining, 2019.

F. Lamraoui and P. Langlais. Yet another fast, robust and open source sentence aligner. time to reconsider sentence alignment. *XIV Machine Translation Summit*, 2013.

P. Langlais, M. Simard, and J. Véronis. Methods and Practical Issues in Evaluating Alignment Techniques. In *36th Annual Meeting of the Association for Computational Linguistics (ACL) and 17th International Conference on Computational Linguistic (COLING)*, pages 711–717, Montreal, Canada, Aug. 1998. doi: 10.3115/980845.980964. URL https://www.aclweb.org/anthology/P98-1117.

S. Laviosa-Braithwaite. Universals of translation. in: Baker, m. (ed.) routledge encyclopedia of translation. london: Routledge. page 288 – 291, 1998.

H. Le, L. Vial, J. Frej, V. Segonne, M. Coavoux, B. Lecouteux, A. Allauzen, B. Crabbé, L. Besacier, and D. Schwab. FlauBERT: Unsupervised language model pre-training for French. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 2479–2490, Marseille, France, May 2020. European Language Resources Association. ISBN 979-10-95546-34-4. URL https://www.aclweb.org/anthology/2020.lrec-1.302.

Y. Li, R. Wang, and H. Zhao. A machine learning method to distinguish machine translation from human translation. In *Proceedings of the 29th Pacific Asia Conference on Language, Information and Computation: Posters*, pages 354–360, Shanghai, China, Oct. 2015a. URL https://www.aclweb.org/anthology/Y15-2041.

Y. Li, R. Wang, and H. Zhao. A machine learning method to distinguish machine translation from human translation. In *Proceedings of the 29th Pacific Asia Conference on Language, Information and Computation: Posters*, pages 354–360, Shanghai, China, Oct. 2015b. URL https://www.aclweb.org/anthology/Y15-2041.

C.-k. Lo. YiSi - a unified semantic MT quality evaluation and estimation metric for languages with different levels of available resources. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 507–513, Florence, Italy, Aug. 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-5358. URL https://www.aclweb.org/anthology/W19-5358.

C.-k. Lo, M. Simard, D. Stewart, S. Larkin, C. Goutte, and P. Littell. Accurate semantic textual similarity for cleaning noisy parallel corpora using semantic machine translation evaluation metric: The NRC supervised submissions to the parallel corpus filtering task. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 908–916, Belgium, Brussels, Oct. 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-6481. URL https://www.aclweb.org/anthology/W18-6481.

A. Lopez. Statistical machine translation. *ACM Computing Surveys*, 40(3), Aug. 2008. ISSN 0360-0300. doi: 10.1145/1380584.1380586. URL https://doi.org/10.1145/1380584.1380586.

J. Lu, X. Lv, Y. Shi, and B. Chen. Alibaba submission to the WMT18 parallel corpus filtering task. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 917–922, Belgium, Brussels, Oct. 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-6481. URL https://www.aclweb.org/anthology/W18-6482.

T. Luong, H. Pham, and C. D. Manning. Bilingual word representations with monolingual quality in mind. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 151–159, Denver, Colorado, June 2015. Association for Computational Linguistics. doi: 10.3115/v1/W15-1521. URL https://www.aclweb.org/anthology/W15-1521.

E. Macklovitch. Using bi-textual alignment for translation validation: the transcheck system. In *First Conference of the Association for Machine Translation in the Americas (AMTA-94)*, Columbia, É-U, oct 1994.

K. Malmkjaer. Norms and nature in translation studies. *Incorporating Corpora - Corpora and the Translator*, pages 49–59, 01 2008.

D. Marcu and W. Wong. A phrase-based, joint probability model for statistical machine translation. In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10*, EMNLP '02, page 133–139, USA, 2002. Association for Computational Linguistics. doi: 10.3115/1118693.1118711. URL https://doi.org/10.3115/1118693.1118711.

L. Martin, B. Muller, P. J. O. Suárez, Y. Dupont, L. Romary, Éric Villemonte de la Clergerie, D. Seddah, and B. Sagot. CamemBERT: a Tasty French Language Model. arXiv 1911.03894, 2019.

T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality, 2013a.

T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc., 2013b. URL http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality pdf.

R. C. Moore. Fast and accurate sentence alignment of bilingual corpora. In *Proceedings of the 5th Conference of the Association for Machine Translation in the Americas on Machine Translation: From Research to Real Users*, AMTA '02, pages 135–144, 2002.

D. S. Munteanu and D. Marcu. Improving machine translation performance by exploiting non-parallel corpora. *Computational Linguistics*, 31(4):477–504, 2005a. doi: 10.1162/089120105775299168. URL https://www.aclweb.org/anthology/J05-4003.

D. S. Munteanu and D. Marcu. Improving machine translation performance by exploiting non-parallel corpora. *Computational Linguistics*, 31(4):477–504, 2005b.

D. S. Munteanu, A. Fraser, and D. Marcu. Improved machine translation performance via parallel sentence extraction from comparable corpora. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 265–272, Boston, Massachusetts, USA, May 2 - May 7 2004. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/N04-1034.

N. Nahata, T. Nayak, S. Pal, and S. Naskar. Rule based classifier for translation memory cleaning. 05 2016.

V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In J. Fürnkranz and T. Joachims, editors, *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 807–814, 2010.

H. Nguyen-Son, N. T. Tieu, H. H. Nguyen, J. Yamagishi, and I. E. Zen. Identifying computer-generated text using statistical analysis. In *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 1504–1511, Dec 2017. doi: 10.1109/APSIPA.2017.8282270.

H.-Q. Nguyen-Son, N.-D. T. Tieu, H. H. Nguyen, J. Yamagishi, and I. Echizen. Identifying computer-translated paragraphs using coherence features, 2018.

H.-Q. Nguyen-Son, T. T. Phuong, S. Hidano, and S. Kiyomoto. Identifying adversarial sentences by analyzing text complexity. *ArXiv*, abs/1912.08981, 2019a.

H.-Q. Nguyen-Son, T. P. Thao, S. Hidano, and S. Kiyomoto. Detecting machine-translated paragraphs by matching similar words, 2019b.

H.-Q. Nguyen-Son, T. P. Thao, S. Hidano, and S. Kiyomoto. Detecting machine-translated paragraphs by matching similar words. arXiv 1904.10641, 2019c.

J.-Y. Nie and J. Cai. Filtering noisy parallel corpora of web pages. *2001 IEEE International Conference on Systems, Man and Cybernetics. e-Systems and e-Man for Cybernetics in Cyberspace (Cat.No.01CH37236)*, 1:453–458 vol.1, 2001.

P. J. Ortiz Suárez, B. Sagot, and L. Romary. Asynchronous Pipeline for Processing Huge Corpora on Medium to Low Resource Infrastructures. In *7th Workshop on the Challenges in the Management of Large Corpora (CMLC-7)*, Cardiff, United Kingdom, July 2019. URL `https://hal.inria.fr/hal-02148693`.

M. Ott, S. Edunov, D. Grangier, and M. Auli. Scaling neural machine translation. *CoRR*, abs/1806.00187, 2018a. URL `http://arxiv.org/abs/1806.00187`.

M. Ott, S. Edunov, D. Grangier, and M. Auli. Scaling neural machine translation, 2018b.

M. Ott, S. Edunov, A. Baevski, A. Fan, S. Gross, N. Ng, D. Grangier, and M. Auli. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-4009. URL `https://www.aclweb.org/anthology/N19-4009`.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

J. W. Pennebaker, M. E. Francis, and R. J. Booth. *Linguistic Inquiry and Word Count*. Lawerence Erlbaum Associates, Mahwah, NJ, 2001.

J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–1543, 2014.

T. Puurtinen. Explicitation of clausal relations: A corpus-based analysis of clause connectives in translated and non-translated finnish children's literature. *Translation Universals: Do They Exist?*, pages 165–176, 01 2004.

A. Pym. 2008.

S. Rarrick, C. Quirk, and W. Lewis. Mt detection in web-scraped parallel corpora. In *Proceedings of MT Summit XIII*. Asia-Pacific Association for Machine Translation, September 2011. URL `https://www.microsoft.com/en-us/research/publication/mt-detection-in-web-scraped-parallel-corpora/`.

N. Rossenbach, J. Rosendahl, Y. Kim, M. Graça, A. Gokrani, and H. Ney. The RWTH aachen university filtering system for the WMT 2018 parallel corpus filtering task. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 946–954, Belgium, Brussels, Oct. 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-6487. URL `https://www.aclweb.org/anthology/W18-6487`.

S. Ruder, A. Søgaard, and I. Vulić. Unsupervised cross-lingual representation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pages 31–38, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-4007. URL `https://www.aclweb.org/anthology/P19-4007`.

G. Saldanha. Accounting for the exception to the norm: Split infinities in translated english. *Language Matters - LANG MATTERS*, 35:39–53, 05 2008. doi: 10.1080/10228190408566203.

G. Sampson. A stochastic approach to parsing. pages 151–155, 01 1986. doi: 10.3115/991365.991407.

V. M. Sánchez-Cartagena, M. Bañón, S. Ortiz-Rojas, and G. Ramírez. Prompsit's submission to WMT 2018 parallel corpus filtering shared task. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 955–962, Belgium, Brussels, Oct. 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-6488. URL `https://www.aclweb.org/anthology/W18-6488`.

R. Sennrich, B. Haddow, and A. Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany, Aug. 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1162. URL `https://www.aclweb.org/anthology/P16-1162`.

R. A. Sharman, F. Jelinek, and R. Mercer. Generating a grammar for statistical training. In *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27,1990*, 1990. URL `https://www.aclweb.org/anthology/H90-1054`.

M. Simard. The BAF: a corpus of english-french bitext. In *First International Conference on Language Resources and Evaluation*, pages 489–494, 1998.

M. Simard. Clean data for training statistical MT: The case of MT contamination. In *Proceedings of the 11th Conference of the Association for Machine Translation in the Americas (AMTA)*, 2014.

M. Simard, G. Foster, and P. Isabelle. Using cognates to align sentences in bilingual corpora. In *Proceedings of the Fourth International Conference on Theoretical and Methodological Issues in Machine Translation*, pages 67–81, 1992.

A. Søgaard, Ž. Agić, H. M. Alonso, B. Plank, B. Bohnet, and A. Johannsen. Inverted indexing for cross-lingual NLP. In *The 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference of the Asian Federation of Natural Language Processing (ACL-IJCNLP 2015)*, 2015.

H. Somers, F. Gaspari, A. Niño, and M. M. Qd. Detecting inappropriate use of free online machine translation by language students – a special case of plagiarism detection. In *Proceedings of the Eleventh Annual Conference of the European Association for Machine Translation*, pages 41–48, 2006.

spaCy. Industrial-strength natural language processing in python. `https://spacy.io`, 2017. (accessed 24-jun-2020).

N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014. URL `http://jmlr.org/papers/v15/srivastava14a.html`.

S. Steding. Machine translation in the german classroom: Detection, reaction, prevention. *Die Unterrichtspraxis/Teaching German*, 42:178 – 189, 11 2009. doi: 10.1111/j.1756-1221.2009.00052.x.

R. Steinberger, A. Eisele, S. Klocek, S. Pilos, and P. Schlüter. DGT-TM: A freely available translation memory in 22 languages. *arXiv preprint arXiv:1309.5226*, 2013.

I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3104–3112. Curran Associates, Inc., 2014. URL `http://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf`.

K. Taghipour and S. Khadivi. Parallel corpus refinement as an outlier detection algorithm. 2011.

C. Teixeira and S. O'Brien. Investigating the cognitive ergonomic aspects of translation tools in a workplace setting. *Translation Spaces*, 6:79–103, 10 2017. doi: 10.1075/ts.6.1.05tei.

B. Thompson and P. Koehn. Vecalign: Improved sentence alignment in linear time and space. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1342–1348, Hong Kong, China, Nov. 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1136. URL `https://www.aclweb.org/anthology/D19-1136`.

J. Tiedemann. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214–2218, Istanbul, Turkey, May 2012. European Language Resources Association (ELRA). URL `http://www.lrec-conf.org/proceedings/lrec2012/pdf/463_Paper.pdf`.

C. Tillmann. A beam-search extraction algorithm for comparable data. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, ACLShort '09, page 225–228, USA, 2009. Association for Computational Linguistics.

G. Toury. In search of a theory of translation, 1980.

G. Toury. Probabilistic explanations in translation studies. in: Mauranen, a., kujamaki, p. (eds) translation universals: Do they exist? amsterdam: Benjamins. page 15 – 32, 2004.

M. Trombetti. Creating the world's largest translation memory. In *MT Summit*, 2009.

D. Varga, P. Halácsy, A. Kornai, V. Nagy, L. Németh, and V. Trón. Parallel corpora for medium density languages. *Amsterdam Studies In The Theory And History Of Linguistic Science Series 4*, 292:247, 2007.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, page 6000–6010, 2017. ISBN 9781510860964.

V. Volansky, N. Ordan, and S. Wintner. On the features of translationese. *Digit. Scholarsh. Humanit.*, 30: 98–118, 2015.

W. Wang, T. Watanabe, M. Hughes, T. Nakagawa, and C. Chelba. Denoising neural machine translation training with trusted data and online data selection. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 133–143, Brussels, Belgium, Oct. 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-6314. URL `https://www.aclweb.org/anthology/W18-6314`.

Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, Łukasz Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, and J. Dean. Google's neural machine translation system: Bridging the gap between human and machine translation, 2016.

H. Xu and P. Koehn. Zipporah: a fast and scalable data cleaning system for noisy web-crawled parallel corpora. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2945–2950, Copenhagen, Denmark, Sept. 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1319. URL `https://www.aclweb.org/anthology/D17-1319`.

Y. Xu, A. Max, and F. Yvon. Sentence Alignment for Literary Texts. *Linguistic Issues in Language Technology*, 12:1–25, 2015. URL `https://hal.archives-ouvertes.fr/hal-01634995`.

K. Yamada and K. Knight. A syntax-based statistical translation model. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, ACL '01, page 523–530, USA, 2001. Association for Computational Linguistics. doi: 10.3115/1073012.1073079. URL `https://doi.org/10.3115/1073012.1073079`.

K. Young, J. Gwinnup, and L. Schwartz. A taxonomy of weeds: A field guide for corpus curators to winnowing the parallel text harvest. In *Proceedings of the 12th Biennial Conference of the Association for Machine Translation in the Americas (AMTA2016)*, 2016.

M. D. Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

A. Zwahlen, O. Carnal, and S. Läubli. Automatic tm cleaning through mt and pos tagging: Autodesk's submission to the nlp4tm 2016 shared task, 2016.