UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola Tècnica Superior d'Enginyeria
de Telecomunicació de Barcelona

Mitigating Social Biases in Machine Translation using Domain Adaptation techniques

Master Thesis
submitted to the Faculty of the
Escola Tècnica d'Enginyeria de Telecomunicació de Barcelona
Universitat Politècnica de Catalunya
by

Adrián de Jorge Sánchez

In partial fulfillment
of the requirements for the master in
TELECOMMUNICATIONS ENGINEERING

Advisor: Marta Ruiz Costa-Jussà
Barcelona, September 2020

Contents

List of Figures

Listings

List of Tables

Revision history and approval record

Revision	Date	Purpose
0	03/07/2020	Document creation
1	03/08/2020	Document revision
2	16/08/2020	Document revision
3	1/09/2020	Document revision

DOCUMENT DISTRIBUTION LIST

Name	e-mail
Adrián de Jorge Sánchez	adjsanchez-@hotmail.com
Marta Ruiz Costa-Jussà	marta.ruiz@upc.edu

Written by:		Reviewed and approved by:	
Date	30/08/2020	Date	01/09/2020
Name	Adrián de jorge Sánchez	Name	Marta Ruiz Costa-Jussà
Position	Project Author	Position	Project Supervisor

Abstract

Misrepresentation of certain communities in current datasets is causing serious disruptions in artificial intelligence applications. Examples of this can be found from lower performance of speech recognizers for women than for men to lower accuracy in face recognition for Asian faces compared to American or European ones. It also amplifies stereotypes in Machine Translation. These challenges are at the core of natural language processing applications and, in particular, there are many works focusing on trying to solve gender biases.

Previous research in the area of Machine Translation (MT) has proposed to either mitigate biases by means of using debiased word embeddings and using contextual information or evaluating and measuring the amount of bias present in the translation. The closest work to ours is one where authors generate a very small gender-balanced dataset and use techniques of Elastic Weight Consolidation to perform transfer learning and mitigate the consequences of training with unbalanced datasets. Differently from this one, we use a larger non-synthetic balanced dataset to perform fine-tunning on an unbalanced-dataset and evaluate the reduction of presence of gender bias in the final translation. We also evaluate the gender bias in word embedding models like in other works, and conclude that they can be successfully applied to downstream systems in the case of the gender-balanced dataset.

Results show that the model that eliminates the gender bias to a greater degree is the model that was fine-tuned with the balanced dataset mixed with a percentage of the original training. This is due to the known difficulties that translation models have when adapting to a new and totally different distribution of data, i.e. catastrophic forgetting, which means that the model fits the new distribution but forgets the one which was trained on before. Some regularization techniques like dropout or adaptive learning rate have been applied, without having a significant improvement. Nevertheless, results show that even if the balanced dataset is from a different domain than the training and the test of the NMT system, it does improve the translation quality (up to 2 BLEU points) and it is able to mitigate the gender bias in a significant amount, up to a 12.5% accuracy.

1 Introduction

Machine translation, is a sub-field of computational linguistics that investigates the use of software to translate text or speech from one language to another. Solving this problem with corpus statistical and neural techniques is a rapidly-growing field that is leading to better translations. Machine Translation has proven useful as a tool to assist human translators and, in a very limited number of cases, can even produce output that can be used as it is.

With the development of the technology, automated language conversion is expected to foresee significant improvements in quality and productivity. Moreover, the industry is moving towards the use of Bilingual Evaluation Understudy (BLEU) as a useful measurement of translation quality. BLEU is adopted as a quality measurement standard that lies with a fully automated way for measuring Neural Machine Translation (NMT) quality.

The Statistical Machine Translation (SMT) segment dominated the machine translation market in 2016. The effectiveness of SMT over Rule-Based Machine Translation (RBMT) in terms of cost and time has significantly increased its demand over the past few years. SMTs are easy to build and maintain and can be adapted to multiple language pairs. SMT development reduces costs of human resources; however, high computational costs are involved in the process.

The industry is experiencing a transition from Human Translation to Machine Translation as is an efficient tool to deliver similar linguistic conversion with significantly lower time and cost. This technology is highly adopted among language services providers, who use this service to enhance their output. Furthermore, use of big data for social media data mining to gather information about products and companies is providing growth opportunities to service providers. Such translated data can prove to be vital for marketing decision-making.

This work has been presented to the 2on Workshop on Gender Bias in Natural Language Processing at COLING 2020 in Barcelona on 13th December 2020 [1]

1.1 Motivation

In Neural Machine Translation, gender bias has been shown to reduce translation quality, particularly when the target language has grammatical gender. Gender bias is an important problem for Neural Machine Translation (NMT) when dealing with gender-inflected languages. Generally, in NMT datasets, it exists an over-prevalence of some gendered forms in the training data, which usually leads to identifiable and incorrect translations. Concretely, translations are better for sentences containing stereotypical gender roles. For example, sentences with mentions of a stereotypical gender role, such as man - computer programmer are more reliably translated than those refering to a male hairdresser. This leads to a decrease in the systems' performance and exhibits gender roles that are not ethic nor fair.

Therefore, we propose to tackle this problem by trying to debias the machine translation system via domain adaptation techniques, feeding a gender-balanced dataset that has the same samples for both genders. In this way, we will be forcing the system to take the attention away from the gender that accompanies the profession and relocate it to the context words.

[1] https://genderbiasnlp.talp.cat/

1.2 Objectives

As previously mentioned, Machine Translation systems are currently very powerful and there is a true competition on having the best system performance, as many services depend on them. We want systems that do not have gender biases (badly associated article, name or adjective, or even verb declinations) when it comes to translating, and therefore, improve the quality of the translation at that level. The objectives of this thesis are the following:

1. Create a gender-balanced dataset.
2. Train a recurrent neural network to be used as a baseline that reaches state of the art performance.
3. Use the gender-balanced dataset to finetune the previously trained network.
4. Analyze gender bias in word embeddings models created independently with both datasets and make a comparison between them, assessing the value of said bias.
5. Analyze gender bias in all translation models and make a comparison between them, trying to reach the conclusion that the model trained with data from the gender-balanced dataset is the least biased.

1.3 Requirements

To achieve all the objectives previously described, certain requirements must be taken into account. These are the applications, source code and data used in this thesis:

- Own source code[2], to perform the gender bias analysis on word embeddings, generate plots and corpus formatting.
- Data from the European Parliament (EuroParl corpus), to train the baseline model and reach the state of the art in terms of performance.
- Gebiotoolkit repository [1], to extract the gender balanced corpus. Some modifications have been made to suit the needs of this project, such as extra functions and queries to the wikipedia API.
- MT Gender repository [2], to analyze the gender bias in translation models, both those developed in-house and commercially.
- Fairseq repository [3], a library that provides many state-of-the-art NMT architectures with useful functions such as preprocessing parallel data, training NMT models and generating translations.
- LASER repository [4], a library to calculate and use multilingual sentence embeddings to extract parallel sentences.
- Sklearn PIP package to perform gender bias analysis on word embeddings, specifically, Kmeans and SVM classes.

[2]Code can be found here: `https://github.com/adridjs/thesis2020`

1.4 Chapters Structure

Based on the ideas presented above, the main part of this thesis is to build a Machine Translation system that is less biased in terms of gender than the baseline one. It is not the goal of this thesis to completely eliminate gender bias, as it has been proven that it is deeply rooted in language [5] [6]. Following this frame of study, the thesis is structured in the following way:

- Chapter 2 - State of the art: Review the current state of the area of research to understand the problem, keep up to date with recent advances in that research field and evaluate possible methods to reach the objectives proposed in Section 1.2.
- Chapter 3: Methodology applied and experiments performed in order to solve the problem.
- Chapter 4: Results about the experiments performed in Chapter 3.
- Chapter 5: Conclusions about the results obtained in Chapter 4.

1.5 Gantt Diagram

- **Planning**: Objectives Definition (1), Requirements Definition (2)
- **Process**: State of the Art research (3), Balanced Dataset Generation (4), Gender Bias Evaluation in Word Embeddings (5), Train Baseline (6), Domain Adaptation (7)
- **Results**: Translation Evaluation (8), Gender Bias Evaluation in NMT (9)

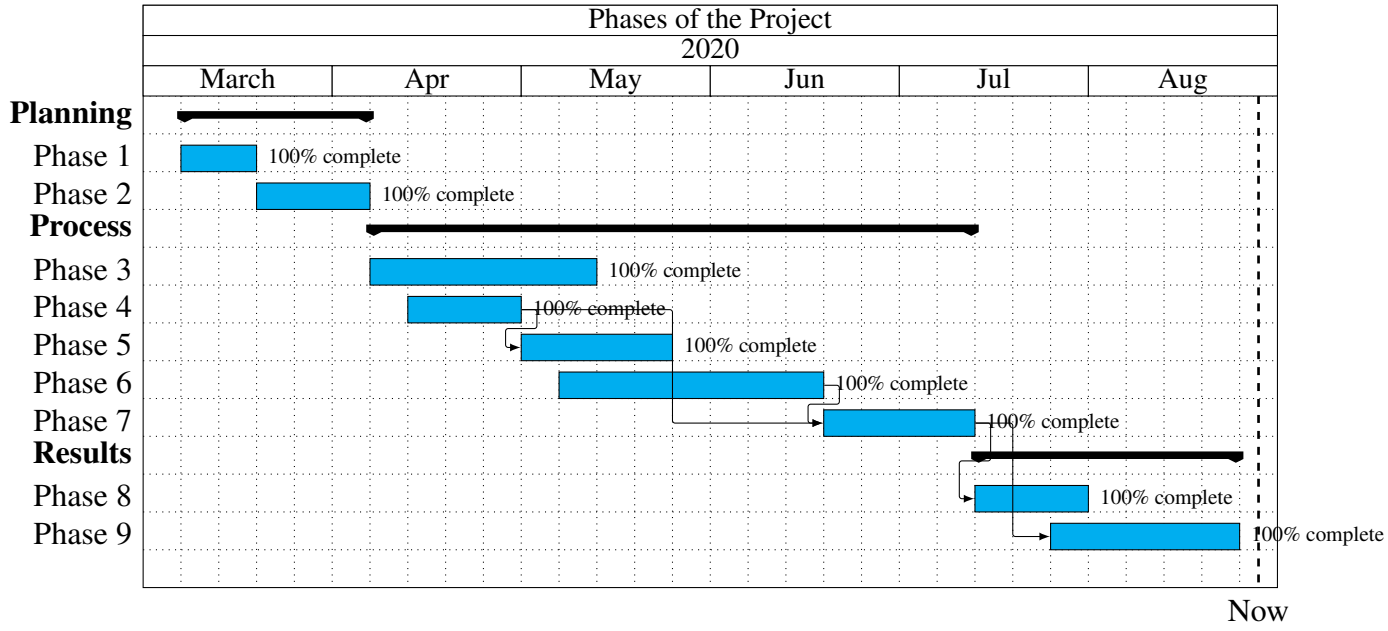


Figure 1: Gantt diagram of the project

2 State of the art

In this section, we will review recent advances of the three fields that concur in the methods that were used in this thesis: Neural Networks for Machine Translation, which is the application of artificial neural networks to predict the likelihood of a sequence of words; Domain Adaptation techniques applied to NMT, which consist of trying to transfer knowledge acquired from a source data distribution to a different, but related, target data distribution; and Gender Bias analysis in NMT, which tries to handle the problem of incorrectly translating from gender-neutral words in a source language to gender-specific words in the target language.

2.1 Neural Networks for Machine Translation

Neural Machine Translation (NMT) is a new approach for Statistical Machine Translation (SMT) that was proposed in 2013 [7]. This approach is inspired by the recent trend of deep representational learning, which has impressed the whole word by reaching or even surpassing human-life performance in various tasks: Chinese-English NMT [8]; Reinforcement Learning such as in AlphaGo [9], which was the first computer program to defeat a professional human Go player and world champion; and Image Recognition from ImageNet [10], which is a large image database based on WordNet having hundreds of photographies per node.

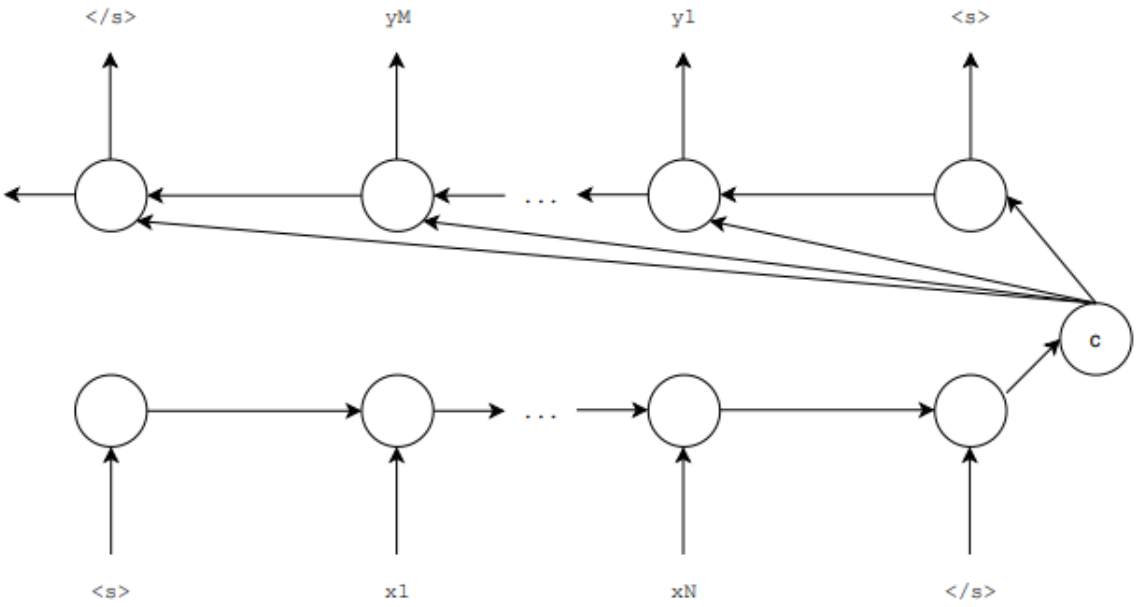


Figure 2: Encoder-Decoder architecture. The encoder tries to condense the data into a lower-dimensional space losing the minimum information. The decoder tries to reconstruct the input from this low-dimensional space.

The neural network models being used in [7], [11] started with an encoder-decoder (see Fig. 2) architecture, in which the former extracted a fixed-length representation from a variable-length input sentence, and the latter generated a translation from this representation.

A RNN works on a variable-length sequence

$$x = (x_1, x_2, \ldots, x_T)$$

by maintaining a hidden state, **h** over time. At each timestep **t**, the hidden state $h(t)$ is updated by

$$h(t) = f(h(t-1), x_t)$$

where f is an activation function. Pretty often, f performs a linear transformation on the input vectors, summing them, and applying an element-wise logistic sigmoid function. An RNN can then be used to learn a distribution over a variable-length sequence by learning the distribution over the next input

$$p(x_{t+1}|x_t, \ldots, x_1)$$

.

2.1.1 Memory in RNNs

A new activation function for RNNs was proposed in [12]. This new function augments the capabilities of the usual sigmoid function

$$\sigma(x) = \frac{1}{1+e^{-x}} = \frac{e^x}{e^x+1}$$

with two gating units called update and reset gates, z and r respectively. Each of these gates depend on the previous hidden state $h(t-1)$, while the current input x_t controls the flow of information. This is similar to the Long Short Term Memory (LSTM) introduced in [13], and tries to combat the vanishing gradient problem by controlling, via the backpropagation signal, the flow of information that goes into the nodes of the network.

The reset gate r_j is defined as

$$r_j = \sigma([W_r x]_j + [U_r h_{t-1}]_j)$$

where σ is the sigmoid function, and the subscript $[.]_j$ denotes the j-th element of a vector. Then, x and h_{t-1} are the input and the previous hidden state, respectively. The matrices, W_r and U_r, are weight matrices that are learned through the backpropagation algorithm [14].

In the same manner, the update gate z_j is defined as

$$z_j = \sigma([W_z x]_j + [U_z h_{t-1}]_j)$$

The activation of h_j is then computed as

$$h_j^t = z_j h_j^{t-1} + (1 - z_j)\widetilde{h}_j^t$$

where

$$\widetilde{h}_j^t = \phi([Wx]_j + [U(r \odot h_{t-1})]_j)]$$

2.1.2 Self-Attention in RNNs

An attention function can be described as a mapping of a query and a set of key-value pairs to an output, where these three are all vectors. The output is computed as a weighted sum of the values, where the weight assigned to each value is computed by a compatibility function of the query with the corresponding key.

Self-attention, is an attention mechanism relating different positions of a single sequence in order to compute a representation of that sequence. This mechanism has been used successfully in a variety of tasks including reading comprehension [15], abstractive summarization [16] and learning task-independent sentence representations [17]

The first step when applying self-attention is to obtain three vectors from each of the encoder's input vectors. For each word, a query q, key k and value v vector is created. These vectors are created by multiplying the input vector by three matrices (W_Q, W_K and W_V) that were obtained during the training process.

Mathematically, having an input sequence of length N represented as a sequence of words

$$\mathbf{x} = [x_0, \ldots, x_{n-1}]$$

In order to compute the self-attention score for each word in the input sentence, we take the dot product of the query vector with the key vector of each of the input words' key vector. For example, if we want to compute the score for the word in the first position (x_0) of the sequence, the scores would be

$$\mathbf{z}_0 = [q_0 k_0, \ldots, q_0 k_{n-1}]$$

After having a vector with the scores, $\mathbf{z}$, these scores need to be divided by the square root of the dimension of the key vectors used, $\sqrt{d_k}$

$$\widetilde{\mathbf{z}} = \frac{\mathbf{z}}{\sqrt{d_k}}$$

Then, the result has to be passed through a softmax operation. This operation has useful properties: normalizes the scores so they're all positive and add up to 1, mimicking probability distributions.

$$\sigma(\widetilde{\mathbf{z}}_i) = \frac{e^{z_i}}{\sum_{j=0}^{N-1} e^{z_j}} = \mathbf{t}$$

Each position of $\mathbf{t}$ will have the softmax score for the word in that position.

The last step is to multiply each value vector by the softmax score, and then summing up the result. This follows the intuition of keeping intact the values of the words that the model is going to focus on, and reducing the amount of signal that flows into the next layer for irrelevant words.

$$\mathbf{a} = [sum(t_0 v_0), \ldots, sum(t_{n-1} v_{n-1})]$$

This vector **a** will hold the self-attention scores for the first word of the sequence. This algorithm needs to be repeated with each of the input sequence's words, where the only change compared to the previous example is the definition of **z**, where, for a word in position i,

$$\mathbf{z}_i = [q_i k_0, \ldots, q_i k_{n-1}]$$

2.1.3 Transformers

The Transformer [18] is a deep learning model introduced in 2017. This is the architecture that was used in this thesis to train a translation model for a given (source-target) language pair. This architecture, unlike RNNs, does not need to process sequential data in order. For example, if the input data is a natural language sentence, the Transformer does not need to process the beginning of it before the end. Thus, due to this feature, the Transformer allows for much more parallelization than RNNs and therefore, its training time is reduced drastically.

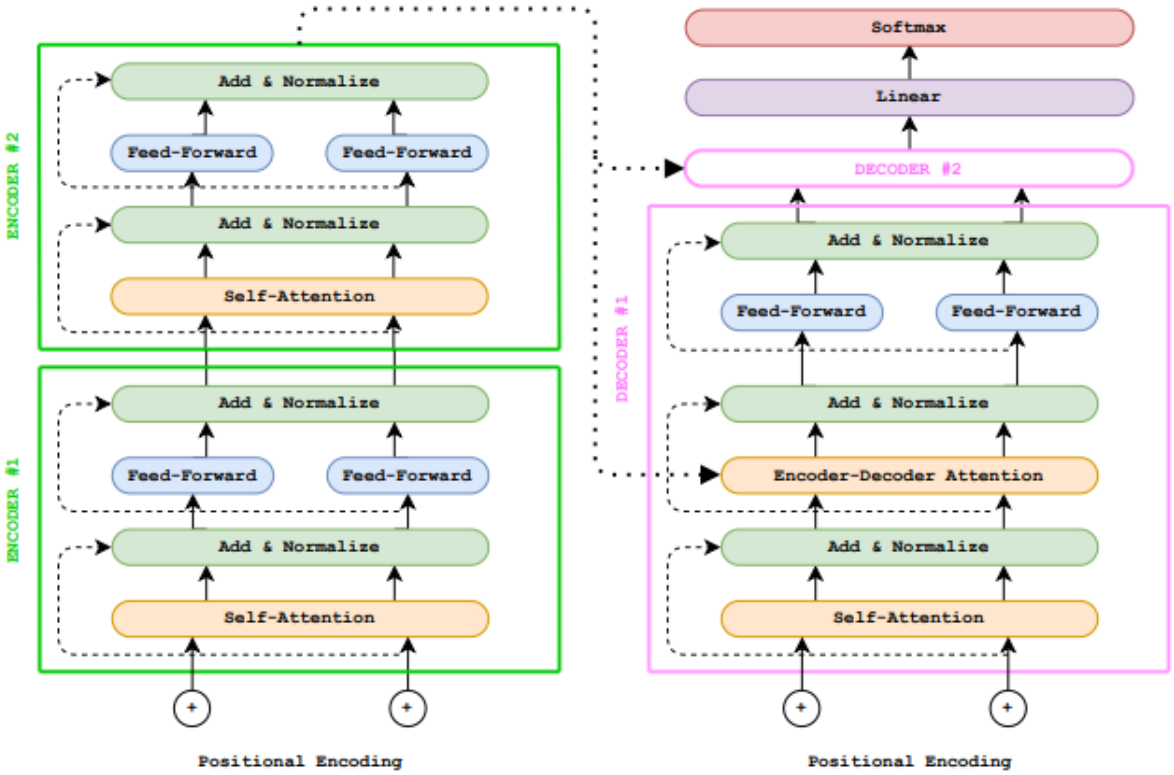


Figure 3: The Transformer architecture, with self-attention layers in each first layer of the encoders and one attention layer for each encoder-decoder connection. There is also a residual connection that passes information to the Add Normalize layer before each feed-forward pass.

Since their introduction, Transformers have become the model of choice for tackling many problems in NLP, replacing older recurrent neural network models such as the long short-term mem-

ory (LSTM) [13]. Since the Transformer model facilitates more parallelization during training, it has enabled training on larger datasets than was possible before it was introduced.

This has led to the development of pretrained systems such as BERT (Bidirectional Encoder Representations from Transformers) [19] and GPT (Generative Pre-trained Transformer) [20], which have been trained with huge general language datasets, and can be fine-tuned to specific language tasks.

The function of each encoder layer is to process its input to generate encodings, containing information about which parts of the inputs are relevant to each other. It passes its set of encodings to the next encoder layer as inputs.

Each decoder layer does the opposite, taking all the encodings and processing them, using their contextual information to generate an output sequence. To achieve this, each encoder and decoder layer makes use of an attention mechanism, which for each input, weights the relevance of every other input and draws information from them accordingly to produce the output.

Each decoder layer has an additional attention mechanism too, which gets information from the outputs of previous decoders. Both the encoder and decoder layers have a feed-forward neural network for additional processing of the outputs, and contain residual connections and layer normalization steps.

2.2 Domain Adaptation in Machine Translation

High quality domain-specific MT systems are in high demand. Due to its limited applications and its usually poor performance, it is important to develop translation systems for specific domains [21].

Leveraging out-of-domain parallel corpora and in-domain monolingual corpora to improve in-domain translation is known as domain adaptation for MT [22, 23]. MT systems typically perform worse in a resource poor or domain mismatching scenario. Hence, it is important to leverage the spoken language domain data with the parent domain data. Moreover, there are monolingual corpora containing millions of sentences for the spoken language domain, which can also be incorporated to this leveraging. [24]

There are many studies of domain adaptation for SMT, which can be mainly divided into two categories: *data centric* and *model centric*. Data centric methods focus on either selecting training data from out-of-domain parallel corpora based on a language model (LM) [25–27] or generating pseudo parallel sentences [28–31]. Model centric methods interpolate in-domain and out-of-domain models in either a model level [26, 32, 33] or an instance level [34]. However, due to the different characteristics of SMT and NMT systems, methods that were originally developed for SMT cannot be directly applied to NMT systems.

2.2.1 Data Centric

Monolingual Corpora In-domain monolingual data cannot be directly used as a Language Model for conventional NMT, unlike in SMT. Many studies have been conducted for this: In this paper [35] propose using target monolingual data for the decoder with LM and NMT multitask learning. [36] use source side monolingual data to strengthen the NMT encoder via multitask

learning for predicting both source sentences reordering and translation. [37] use both source and target monolingual data for NMT through reconstructing the monolingual data by using NMT as an autoencoder.

Out-of-Domain Parallel Corpora With both in-domain and out-of-domain parallel corpora, it is ideal to train a mixed domain MT system that can improve in-domain translation while do not decrease the quality of out-of-domain translation. These are categorized as multi-domain methods, that have been successfully applied to NMT.

The multi-domain method in [38] is originally motivated by [39]. In this method, the corpora of multiple domains are concatenated with two small modifications:

1. Appending a domain tag to the source sentences of the respective corpora. This makes the NMT decoder to prioritize generating sentences for that specific domain.
2. Oversampling the smaller corpus so that the training procedure pays equal attention to each domain.

In this paper, [40] they compare different methods for training a multi-domain system. They find that fine tuning on the concatenated multi-domain corpora shows the best performance.

2.2.2 Model Centric

Training Objective The methods in this section change the training functions or procedures for obtaining an optimal in-domain training objective.

For *Cost Weighting*, the NMT cost function is modified with a domain classifier [41]. The output probability of the domain classifier is transferred into the domain weight. This classifier is trained using development data.

In *Fine Tuning* methods, which are the conventional way for domain adaptation, a NMT system is trained on a rich resource of out-of-domain corpus until convergence, and then its parameters are fine tuned on a resource poor in-domain corpus.

Conventionally, fine tuning is applied on in-domain parallel corpora. To prevent performance degradation of out-of-domain translation after fine tuning on in-domain data, [42] propose an extension of fine tuning that keeps the distribution of the out-of-domain model based on knowledge distillation [43]. Knowledge distillation is the process of transferring knowledge from a large model with high knowledge capacity - even if it is not fully used - to a smaller one without losing performance.

In *Mixed Fine Tuning*, they combine *multi-domain* and *fine tuning* approaches. The training procedure is the following:

1. Train an NMT model on out-of-domain data until convergence.
2. Resume training the NMT model from step 1 on a mix of in-domain and out-of-domain data until convergence.

Mixed fine tuning addresses the over-fitting problem of fine tuning due to the small size of the in-domain data by oversampling this in-domain data. It is easier to train a good model with

out-of-domain data, compared to training a multi-domain model.

Once the parameters for the out-of-domain data are obtained, we can use these parameters to fine tune on the mixed domain data. In addition, mixed fine tuning is faster than multi-domain because it is faster to train an out-of-domain model - because it convergences faster than a multi-domain model - and it also convergences faster when fine tuning later.

In [38], *mixed fine tuning* is shown to perform better than *multi-domain* and *fine tuning* approaches. In addition, mixed fine tuning has the similar effect as the ensembling method in [42], which does not decrease the out-of-domain translation performance.

Others There are other model centric approaches such as changing the NMT model architecture, training an in-domain RNN-LM for the NMT decoder and combine it with an NMT model [44] or discriminating the domain, by adding a feed-forward network that acts as a discriminator of the source sentence domain, to leverage the diversity of information in multi-domain corpora [45].

2.3 Gender Bias in Machine Translation

Machine translation models are trained on huge corpora of text, consisting of parallel sentences in a (source, target) language pair. These sentences are commonly extracted from news, weblogs or even talk shows. These datasets have been shown to reflect social biases, commonly by the under-representation in them of certain races or genders, when assessing racial or gender bias, respectively.

In many cases, the bias of a NMT system is not caused by an active bias of machine learning developers, but rather by the inherent societal biases that the data sources contain. This bias is then manifested in datasets that are created from these types of sources. For example, if more women than men have historically been nurses, the machine learning model trained on these data will learn that nurses are more likely to be women than men, assuming that distribution still holds nowadays.

Gender bias in Machine Translation tries to detect and correct wrongly translated sentences due to a mismatch on the source and target words that are gendered. This has been shown to reduce translation quality, particularly when the target language has grammatical gender. For example, when translating from English to Spanish, there can be some gender-neutral words in English that are gender-specific in Spanish.

Recent approaches have involved training from scratch on artificially created gender-balanced versions of the original dataset [46, 47], where in the former they use this dataset to evaluate the gender bias in coreference systems, and in the latter, they propose a method to swap genders in sentences of languages with rich morphology by using dependency trees, part-of-speech tags and morpho-syntactic tags from Universal Dependencies.

Debiased word embeddings are also another approach that has been increasing its interest in the recent years. In this paper [48], they use GloVe word embeddings and its debiased counterparts (GN-GloVe, Hard-Debiased GloVe) by tweaking the first layer of the Transformer, which is the one that learns the word representations. Then they compare the BLEU performance to that of

the baseline, concluding that the model trained using the GN-GloVe embeddings reaches the best performance. Regarding the gender bias evaluation, the Hard-Debiased GloVe performs much better than any other system, reaching almost 100% accuracy at co-referent resolution tasks. In this other paper [6], they firstly identify the direction of the embeddings where the bias is present. Then, gender neutral words that have components in this direction are set to zero, equalizing the sets by making the gender-neutral word equidistant to gender-specific words in the given set.

Another approach which has been shown to mitigate gender bias is to treat it as a domain adaptation problem. In this paper [49], They first create a small, hand-crafted dataset with a list of professions from US labour statistics, and then they perform counterfactual data augmentation. This process is commonly used to handle data over-representation. In this case, it consists in finding gendered sentences and switch genders in source and target languages. They compare Elastic Weight Consolidation (EWC) [50] and Lattice Rescoring [51], showing that the latter performs better and with the advantage of not requiring access to the original model.

3 Methodology

In this section, we explain the procedure that has been followed to obtain an English-Spanish gender-balanced dataset, which uses the available Gebiotoolkit [1] and extracts parallel corpus at the level of sentence from the Wikipedia Biographies. Here in after, we refer to this dataset as Balanced. We quantify the amount of gender bias in the collected dataset as a reflection of the amount of gender bias in the words embeddings. This quantification of bias is also compared to the case of words embeddings computed on the EuroParl corpus [52].

3.1 Balanced Dataset Generation

We used the available Gebiotoolkit [1] to extract the Balanced dataset. Gebiotoolkit is a tool for extracting multilingual parallel corpora at sentence level, together with document and gender information from Wikipedia biographies. In this sense, the collected data set is not synthetic. The dataset can be generated from any of the languages available in the Wikipedia, in our case, we have selected the English-Spanish language pair, which have considerable differences at the morphological level, and exhibit gender bias issues in MT [48]. Gebiotoolkit requires three inputs:

1. A list of the desired languages, which were set to English and Spanish.
2. A list of the article titles belonging to the category to extract - which in our case it is 'Living People' - in English.
3. The wikipedia dump[3] files for the languages that were set on (1) in order to extract the articles requested in (2).

.

In order to retrieve the list of articles under an specific category, PetScan tool[4] was used. The corpus extractor module starts by looking for the equivalent articles to those input for the other languages via the Wikipedia interlanguage links.

GeBioToolkit uses a modified version of the wikiextractor [53] software to retrieve and store the different Wikipedia entries from each language. Finally, file selection generates a dictionary similar to the one obtained before, but it only stores the entries for which the files were successfully retrieved.

After the articles extraction step, the corpus alignment module makes use of the information retrieved in the previous step and the LASER toolkit [54]. LASER (Language-Agnostic SEntence Representations) allows to obtain sentence embeddings through a multilingual sentence encoder. This system uses a single BiLSTM (see Fig. 4) encoder with a shared BPE vocabulary for all languages, which is coupled with an auxiliary decoder and trained on publicly available parallel corpora.

Sentences from all languages are mapped into the same embedding space, so embeddings from different languages are comparable. The sentence in the source language is encoded by the

[3] https://dumps.wikimedia.org/
[4] https://petscan.wmflabs.org/

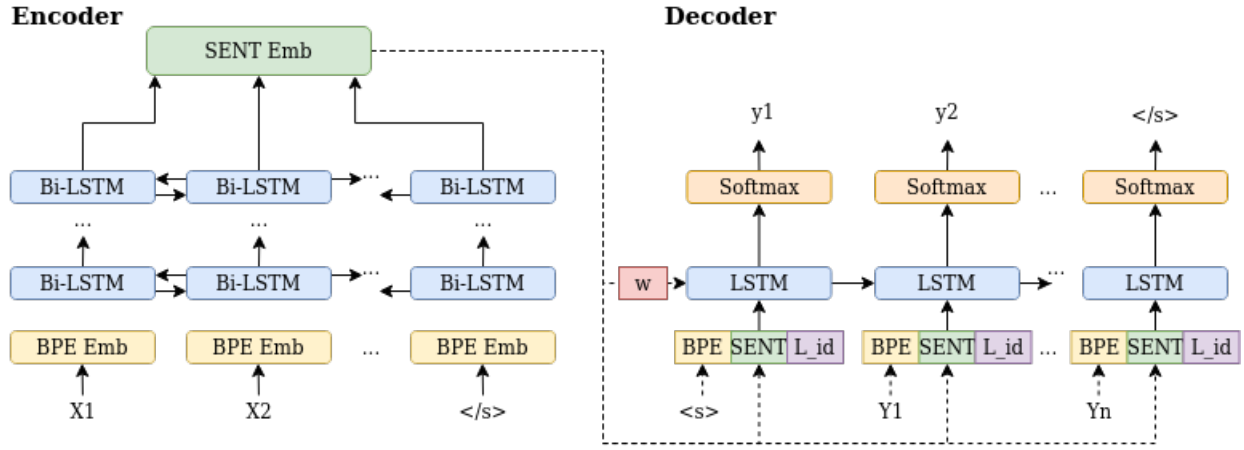


Figure 4: LASER architecture. The encoder does not know the language which is being fed while the decoder has this information appended to its input via the L_{id} tag

encoder, and then translated to the target language by the decoder. The encoder does not know what the target language is, making the system language-agnostic. The target language is given to the decoder through the input signal (see Fig 4). Translations can be found then as close pairs in the multilingual semantic space.

After the alignment, data is stored by language and gender (see Table 1), for which each sentence in every file has its parallel, translated sentence in the same line. The data is cleaned by removing Wikipedia anchor tags .

Aligned Sentences	
en	es
Andrew_Parrott : He was music director of the London Mozart Players for several years until September 2006.	Andrew_Parrott : Posteriormente fue director musical de los London Mozart Players durante varios años hasta septiembre de 2006.
Paul_Heyman : On the November 11, 2013 episode of ”Raw”, Heyman stated that he was no longer with Ryback as Ryback never officially accepted his proposal to become a ”Paul Heyman Guy”.	Paul_Heyman : En el episodio del 11 de noviembre de 2013 de ”Raw”, Heyman declaró que ya no estaba con Ryback ya que Ryback nunca oficialmente había aceptado su propuesta para convertirse en un ”Paul Heyman Guy”.
Richard_Florida : Florida’s earlier work focused on innovation by manufacturers.	Richard_Florida : Los primeros trabajos de Florida se centraron en la innovación industrial.

Table 1: Examples of aligned sentences.

The biographies dataset has approximately 27,000 female-related sentences and 47,000 male-related sentences. These sentences are typically short and some are not perfectly written, In order to have an equal probability of finding a male or female related sentence, the dataset was

balanced by removing male-related samples until having the same amount of sentences between genders.

After extraction, the biographies dataset has approximately 27,000 female-related sentences and 47,000 male-related sentences. In order to have an equal probability of finding a male or female related sentence, we balanced the dataset by removing male-related samples until having the same amount of sentences between genders. In total we end up with 54,000 parallel sentences and the word embedding model has a vocabulary size of 17,277 English words.

Similarly, the Europarl corpus has 2,007,758 parallel sentences and its word embedding model has a vocabulary size of 87,033 English words.

3.2 Gender bias Analysis through Word Embeddings

To evaluate the amount of bias in the Balanced dataset, we build word embeddings which is a vectorization of words following the Word2Vec [55] technique and we assume that the presence of bias in words embeddings is a kind of reflection of the biases in the dataset [56].

We use 128 as the number of dimensions for these vectors, a minimum count of 5 in order to remove poorly represented words and a bidirectional window of 3 words, that is, given a word $x[n]$, its "context" is

$$x[n-3],\ldots,x[n],\ldots,x[n+3]$$

To perform the gender bias analysis of these words embeddings, we use the measures proposed in previous works [5, 6]. Inspired by these previous studies, we make use of the following lists of words:

- Definitional Pairs List
- Biased List, which contains of 1000 words, 500 female biased and 500 male biased. (e.g. diet for female and hero for male)
- Extended Biased List, extended version of Biased List (5000 words, 2500 female biased and 2500 male biased)
- Professional List 319 tokens (e.g. accountant, surgeon)

The definition of gender bias and its evaluation is taken from [6], where they define the gender bias of a word $\overrightarrow{w}$ by its projection on the gender direction, assuming all vectors are normalized

$$\overrightarrow{w} \cdot (\overrightarrow{he} - \overrightarrow{she})$$

3.2.1 Gender Direction and Direct Bias

Following the previous study [6], we took the M gender pair difference vectors

$$\begin{bmatrix} \overrightarrow{w}(he) - \overrightarrow{w}(she) \\ \overrightarrow{w}(father) - \overrightarrow{w}(mother) \\ \cdots \\ \overrightarrow{w}(son) - \overrightarrow{w}(daughter) \end{bmatrix}$$

from the Definitional List (see Table 2) and computed its principal components (PCs) in order to identify the gender subspace. We then generate a random base of M units vectors of 128 dimensions for comparison. Figure 5 shows the PCA plots in both the gendered and the random vectors.

In the case of the EuroParl dataset, there is a clear dominance of one gender direction in the PCA from gender vectors. In the case of the Balanced datasets, the dominance is lower, but we can see that the 2 PCs from the left image (our gender base) explain almost 65% of the variance (information).

he - she	boy - girl
father - mother	male - female
his - her	himself - herself
man - woman	son - daughter

Table 2: Definitional List used in gender bias evaluation.

We take the definition of gender bias from [6], where they define the gender bias of a word $\overrightarrow{w}$ by its projection on the gender direction $\overrightarrow{g}$. The higher the magnitude of the projection onto the previously defined base, the more biased the word is. We use the lists of neutral professions in [57] in order to compute the direct bias of our Balanced dataset as follows.

$$\frac{1}{|N|}\sum_{\omega \epsilon N} |cos(\overrightarrow{\omega} \cdot g)| \tag{1}$$

After filtering by words that exist in our word embeddings model, we get N=147 for the Europarl dataset and N=140 for the Balanced dataset. Direct bias is 0.23 for the EuroParl and 0.10 for the Balanced dataset. This measure confirms that most words still have some of its information alongside the gender direction. These results are higher of what is reported in Bolukbasi's work (although it is not directly comparable). Having a lower N may interfere in the direct bias measure.

3.2.2 Clustering

Are stereotypically-gendered words easy to cluster based on their word embedding representations? K-means is a clustering is a method of vector quantization that aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean (cluster centroid), serving as a prototype of the cluster. In order to perform this clustering, *Scikit* [58] was used, which is a scientific python library that, among many other things, has a K-means implementation.

As the purpose of this experiment was trying to cluster out male and female word embeddings, $k = 2$ was set. Take into account that higher number of clusters can be set in order to gain more insight of word distribution in the embedding space.

The clustering measure wants to evaluate if stereotypically-gendered words (Biased List) are easy to cluster based on their word embedding representations. The higher the clustering accu-

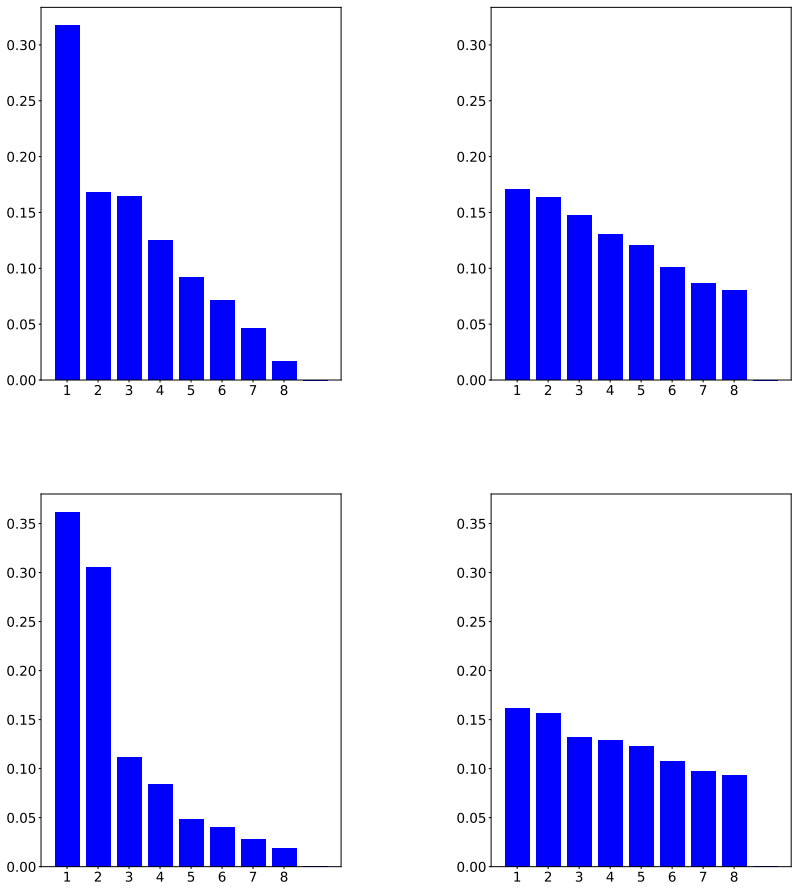


Figure 5: PCA Comparison between the gender base (left) and a randomly generated base (right) of 128 dimensions from Europarl (top) and Balanced (bottom) datasets.

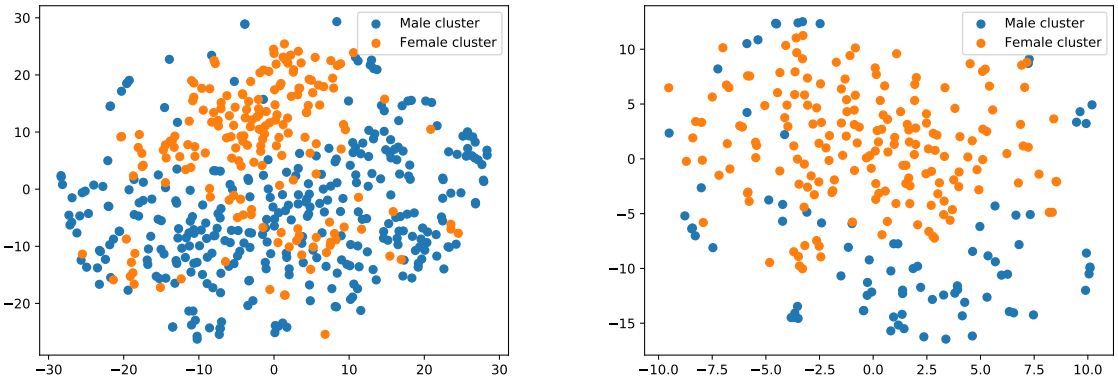
racy, the more bias the words embeddings have. We use *Scikit learn* [58] toolkit to perform an unsupervised k-means clustering classification (with 2 clusters).

Figures 6 and 6b show the tSNE projections of the vectors for both Europarl and Balanced datasets. The clustering model trained with the Europarl aligns with gender with an accuracy of 77.67% and Balanced dataset word embeddings aligns with gender with an accuracy of 68.47%. Note that not all the words in the Biased List appear in our dataset, in fact, we were only able to use 263 words and 512 words[5] (out of 1000) from the original Biased List, respectively,

3.2.3 Classification

We want to know if stereotypically-gendered words (Extended Biased List) can be classified into masculine or feminine based solely on their word embedding representations. We build a RBF-kernel SVM classifier to discover if the model can generalize its predictions into other

[5]Words used can be found in `https://github.com/adridjs/thesis2020/tree/master/gender_bias/data/`



(a) Clustering on EuroParl

(b) Clustering on Balanced

Figure 6: tSNE projection after K-means clustering on Balanced and EuroParl datasets.

stereotypically-gendered words. We perform the evaluation on the Balanced and EuroParl corpus.

We start from the Extended Biased List of the 5000 most-biased words in [5] according to the original bias (2,500 from each gender). We then split these into train and test sets, drawing a 20% (255 words) for the train set and 1022 for testing the performance of the model. The accuracy of the classifier for the Europarl dataset is 80.59% and for the Balanced dataset is 73.28%.

That is a good result as it means that the representation of words as vectors is not a good discriminatory variable to take into account when trying to classify words by gender. This has implications in the downstream systems that are going to be trained with these vectors (if any), as it will incorporate little bias with these representations.

3.2.4 Discussion

The accuracy reported in Europarl and Balanced datasets is not comparable since both have different number of total and vocabulary words. We know that the word embedding representation changes when having more word repetitions. Having said that, results in absolute terms tends to report less bias in the Balanced dataset compared to the Europarl dataset. Moreover, these results are also lower than the ones reported in previous studies [5].

3.3 Use of Domain Adaptation techniques for Gender Bias Mitigation

In this section we use the gender-balanced dataset described in the previous section to mitigate the gender bias present in a standard MT system. We build the Neural MT system using the standard Transformer [18] on a large dataset. Our idea is to use fine-tuning techniques with the balanced dataset on this baseline system.

3.3.1 Training Pipeline

The idea is that we have a parent translation model trained with unbalanced data and we want to learn a child model taking advantage of the balanced dataset. To avoid catastrophic forgetting, where the child model forgets everything that has been learnt from the parent, we use the mix fine tunning strategy. This strategy which consists in initializing the child model with the parent model and train it on a percentage of the unbalanced data set concatenated with the entire balanced data set has been proven to mitigate the catastrophic forgetting problem. [59].

We train the parent model with a large dataset and then fine-tune it with 3 types of datasets: Balanced, a Mix of the Large and Balanced dataset, having different proportions of the large dataset into it, and Concat which contains the entire Large and Balanced datasets (see Figure 7).

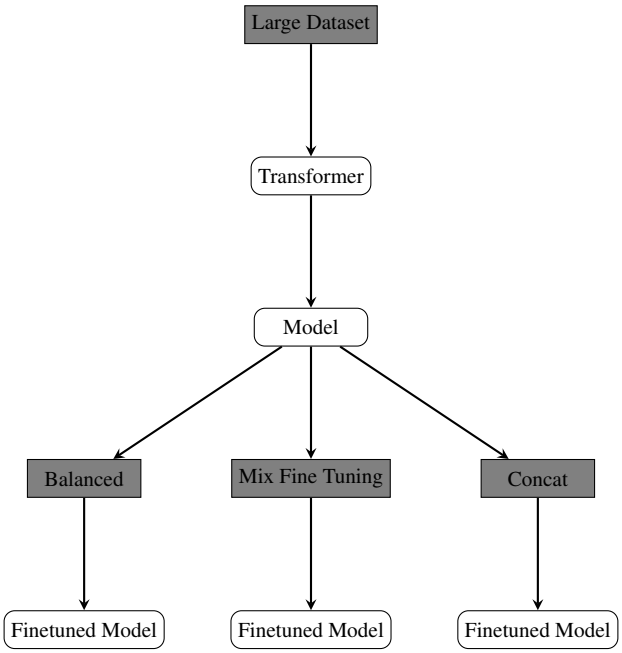


Figure 7: NMT training pipeline. The gray boxes represent the corpus used to train the model.

3.3.2 Experimental Framework

Generic Data To train the parent model, we used the English-Spanish EuroParl corpus [52], which contains parallel data from the proceedings of the European Parliament. We extract a part of the corpus that consists of 2 million parallel sentences. We applied a preprocessing step that consisted of tokenizing, truecasing and filtering. All these steps were performed using scripts from the well-known Moses [60] scripts.

1. The tokenizing step consisted of (A) punctuation normalization and (B) tokenizing the text itself.
2. The truecasing step learns a model (which is a list of words and the frequency of their different forms) from the training partition and only changes the words at the beginning

of the sentence to their most frequent form. This model is later applied to all partitions, i.e, train, test and dev.

3. The filtering step removed parallel sentences with length greater than 50, and sentence pair ratio greater than 9:1.

After the preprocessing step, the corpus size reduces to 1.77 million parallel sentences.

Parameters The network is trained for an undefined number of epochs until convergence with an early stopping policy. That policy consists of setting a *patience*, that is, if the validation loss does not improve in *patience* epochs, stop the training. We set that to 5 as it gives good results empirically. We used 512 embeddings dimention, 6 layers in the encoder and decoder, 8 attention heads. We used a batch size of 16, a dropout of 0.1 and a learning rate of 0.001. We optimized with Adam. For a full list of used parameters, refer to Table 8 in the appendix.

Architecture We use the Transformer [18] as baseline NMT model architecture, an encoder-decoder architecture which is entirely based on attention-based mechanisms that boosts the performance in MT tasks compared to RNNs or CNNs architectures.

3.3.3 Fine-tuning

The Baseline model is fine-tuned using dropout set to 0.3. This is used as a regularization technique together with the mixed fine tuning approach to handle the catastrophic forgetting problem.

Balanced We hypothesize that fine-tuning on a corpus which is balanced in gender will improve the accuracy in gendered translations. We use the corpus extracted by Gebiotoolkit as reported in Section 3.1 - which is balanced in gender - in order to test this hypothesis. Note that the Balanced data is from a different domain than the training and test data.

Mix This approach is building a dataset based on a mix of EuroParl and Balanced datasets. We study the influence in gender bias and MT performance by having more or less in-domain data being fed in the fine-tuning step. More percentage means more EuroParl data.

Concat This approach consists of concatenating the whole EuroParl corpus with the Gender-Balanced biographies dataset.

Note that Balanced and Concat could also lie into the mix fine-tuning strategy, being 0% and 100%, respectively, the percentage of sentences from the Europarl corpus.

4 Results

Findings are presented from Table 3 to 6. Two baseline models are reported: one trained with the EuroParl corpus and another trained with the concatenated dataset (Base-Concat) composed by EuroParl and Gebiotoolkit dataset. An evaluation in terms of translation performance and an evaluation in terms of gender bias accuracy are reported.

We use BLEU [61] to evaluate the performance of our translation models. We use the gender bias evaluation pipeline from [2], also known as WinoMT to evaluate the gender bias in these models.

4.1 Translation Evaluation

Our baseline model is the one trained with the EuroParl corpus. We trained the same Transformer model with a concatenated dataset composed by EuroParl and Gebiotoolkit dataset, which shows an increment of 1.5 points in the english-to-spanish model and almost the same increment in the reversed model.

Translation Performance				
Corpus	Test Set			
	NewsTest2013		Gebiocorpus	
	en2es	es2en	en2es	es2en
EuroParl	26.87	25.50	44.04	40.95
Base-Concat	28.37	26.91	45.82	43.05
FT-Balanced	27.51	27.80	**46.60**	**45.63**
FT-Mix 5%	28.51	28.71	46.17	44.44
FT-Mix 10%	28.52	28.76	46.32	44.10
FT-Mix 20%	28.72	28.78	46.28	44.86
FT-Mix 30%	**28.76**	28.95	46.35	45.12
FT-Mix 40%	28.61	**29.05**	46.43	45.37
FT-Concat	28.68	28.29	46.56	45.52

Table 3: BLEU results for the different trained systems. Bold numbers represent best performance column-wise.

All the fine-tuned models surpass these two in both test sets, except the FT-Balanced in the *en2es* model for the NewsTest2013 set. The best performance achieved on the *en2es* model is the one where 30% of EuroParl data is present, whereas in the *es2en* model, this is achieved by the model that was trained with a 40% of the data from EuroParl.

Regarding the Gebiocorpus test set, which are manually selected and translated sentences from Wikipedia biographies, the best performance in both languages is achieved by the system fine-tuned on the Balanced dataset. This is normal as the sentences from train and test set share the same domain, which makes it easier for the system to generalize.

4.2 Gender Bias Evaluation

The dataset consists of 3888 sentences. In each of these sentences, there is a primary entity which is coreferent with a pronoun, and a secondary entity, that tries to trick the translation system. The scripts provided by the authors extract the grammatical gender of the primary entity from each translation by automatic word alignment and followed by morphological analysis. Then, it compares the translated primary entity with the annotated gender. The objective is to have a translation where the primary entity's gender matches the gold annotated one.

The metrics reported in the gender bias part are the overall system accuracy, *i.e* Acc., which is the percentage of instances in which the translation preserved the gender of the entity from the original sentence, and Δ_g, which tracks the difference between male and female F-scores.

All systems perform quite poor on the accuracy metric, where the best performing model does not achieve better than random guessing in order to get the correct gender inflection. For the Δ_g measure, we can see an strong deviation by the FT-Concat model, which indicates that it is by far the least biased model. Note that all models perform better on male instances than on female ones, caused by the unbalanced distribution between genders in the training set.

4.2.1 General Bias

For the general bias measures, the best performance is achieved with FT-Concat, getting 49.8% accuracy at identifying the correct gender when translating into spanish. It doesn't have the highest F-score for males, but it does for females, and the difference between the scores, Δ_g is much higher in the latter than in the former, thus giving a lesser biased performance, as reported here [2].

General Gender Bias				
Corpus	Acc.	F-Score		Δ_g
		M	F	
EuroParl	46.6%	59.8%	31.3%	28.5
Base-Concat	47.3%	60.3%	32.4%	27.9
FT-Balanced	48.3%	60.4%	33.8%	26.6
FT-Mix 5%	47.5%	60.2%	32.0%	28.2
FT-Mix 10%	47.9%	60.4%	32.6%	27.8
FT-Mix 20%	48.2%	60.7%	33.3%	27.4
FT-Mix 30%	48.8%	60.8%	35.2%	25.6
FT-Mix 40%	49.0%	**61.1%**	35.5%	25.6
FT-Concat	**49.8%**	59.9%	**41.7%**	**18.2**

Table 4: Accuracy in the General WinoMT test set. Bold numbers represent best performance column-wise.

4.2.2 Pro-Stereotypical Bias

In this setup, FT-Concat performs much better than any other model. Its accuracy is 10 points higher than the best baseline model. Its F-score differences are also the lowest, meaning that

there's less bias than in any other trained model.

Pro-stereotypical Gender Bias				
Corpus	Acc.	F-Score		Δ_g
		M	F	
EuroParl	53.5%	67.7%	35.9%	31.8
Base-Concat	56.2%	69.1%	38.8%	30.3
FT-Balanced	59.3%	70.0%	47.7%	22.3
FT-Mix 5%	57.3%	69.3%	43.2%	26.1
FT-Mix 10%	57.8%	69.4%	44.1%	25.3
FT-Mix 20%	58.2%	69.9%	44.6%	25.3
FT-Mix 30%	58.9%	70.3%	46.0%	24.3
FT-Mix 40%	59.0%	70.8%	45.5%	25.3
FT-Concat	**66.3%**	**74.1%**	**62.0%**	**12.1**

Table 5: Accuracy in the WinoMT test set. Pro-Stereotypical translations. Bold numbers represent best performance column-wise.

4.2.3 Anti-Stereotypical Bias

Lastly, the best model performance is obtained on the FT-Mix40% model, which has an accuracy of 45% (lowest for all the setups). The minimum F-score difference is 28'9%, which is very high. In general, we can see that in this setup the models do not perform very well. This reveals that the systems are still biased, as we have low anti-stereotypical and high pro-stereotypical translation performance.

Anti-stereotypical Gender Bias				
Corpus	Acc,	F-Score		Δ_g
		M	F	
EuroParl	44.3%	57.1%	28.2%	28.9
Base Concat	39.0%	52.3%	21.5%	30.8
FT-Balanced	43.1%	56.7%	22.9%	33.8
FT-Mix 5%	43.1%	56.6%	23.5%	33.1
FT-Mix 10%	43.4%	57.1%	23.2%	33.9
FT-Mix 20%	44.1%	**57.4%**	24.7%	32.7
FT-Mix 30%	44.3%	57.1%	26.6%	30.5
FT-Mix 40%	**45.0%**	**57.4%**	**28.6%**	**28.8**
FT-Concat	44.5%	57.0%	26.3%	30.7

Table 6: Accuracy in the WinoMT test set. Anti-Stereotypical translations. Bold numbers represent best performance column-wise.

4.3 Generated Translations

Some of the translations generated by the model with best performance in terms of gender bias and an explanation of the phenomena ocurrying in each case can be seen in the following Table 7. The sentences are taken from WinoMT test set, which were used to compute the gender bias of the NMT system.

Source Language (en)	Predicted Translation (es)	Case
The lawyer told *the designer* that she may lose **her** job due to the prosecution.	El abogado le dijo a *la diseñadora* que podía perder su trabajo debido a la acusación.	Good coreference resolution with a female stereotypical profession.
The editor asked *the guard* to leave the room and asked **her** to say nothing.	El editor pidió a *la guardia* que abandonara la sala y le pidió que no dijera nada.	Good coreference resolution with a female antistereotypical profession.
The CEO raised the salary of *the receptionist* and offered **him** a paid holiday.	El director ejecutivo elevó el salario *del recepcionista* y le ofreció unas vacaciones remuneradas.	Good coreference resolution with a male antistereotypical profession.
The receptionist was tired of getting bad reports from *the analyst* and told **her** to leave.	El recepcionista estaba cansado de recibir informes negativos *del analista* y le dijo que se marchara.	Bad coreference resolution. The system outputs *del analista*, which is masculine-gendered.

Table 7: Examples of translated sentences by the FT-Concat model. The cursive words represent the entity which the word in bold is refering to.

5 Conclusions

The experiments presented in the previous sections have revealed a systematic bias in MT systems and - in a lesser degree - in word embeddings models.

Our first hypothesis was that giving a closed, in-domain, gender-balanced dataset would diminish the gender bias in NMT systems. It has been proven that it does in word embeddings representations, meaning that these representations could be used by downstreams systems with the ability of having little to no gender bias.

On the other hand, from these results, providing an small, gender-balanced dataset does not improve performance on NMT systems regarding gender bias metrics. The best performance is achieved on the fully concatenated *EuroParl + Balanced* dataset, which arises two questions: Is this a problem on the difference of corpus size between both datasets? Is this a problem of the type of fine-tuning approach that was taken?

It is indeed clear that more data implies more richness in its probability distribution, as it is more probable to find extreme cases or cases that did not appear in a subset of said data. Getting gender-balanced datasets which are not synthetic and large enough to be compared to the existing parallel datasets is difficult, thus making the task of gender debiasing on the training step a challenge: you either adapt the model too much to the new data distribution, which is more compact and constrained, making the model lose its ability of generalization on more common translations, or you restrict the update of the model parameters and do not debias at all, which was the primary task to be pursued.

Translation quality was improved up to 2 BLEU points and gender bias was mitigated by a significant amount, up to a 12.5% accuracy with the FT-Concat model. We think that there's still a lot to improve regarding gender bias in MT systems, as the anti-stereotypical translation performance - which plays an important role in measuring gender bias - was considerably lower than the other two setups.

References

[1] Marta R. Costa-jussà, Pau Li Lin, and Cristina España-Bonet. Gebiotoolkit: Automatic extraction of gender-balanced multilingual corpus of wikipedia biographies. In *Proceedings of 12th Language Resources and Evaluation Conference (LREC)*, 2019.

[2] Gabriel Stanovsky, Noah A. Smith, and Luke Zettlemoyer. Evaluating gender bias in machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1679–1684, Florence, Italy, July 2019. Association for Computational Linguistics.

[3] Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*, 2019.

[4] Holger Schwenk and Matthijs Douze. Learning joint multilingual sentence representations with neural machine translation. In *Proc. of the 2nd Workshop on Representation Learning for NLP*, pages 157–167, 2017.

[5] Hila Gonen and Yoav Goldberg. Lipstick on a pig: Debiasing methods cover up systematic gender biases in word embeddings but do not remove them. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *NAACL-HLT (1)*, pages 609–614. Association for Computational Linguistics, 2019.

[6] Tolga Bolukbasi, Kai-Wei Chang, James Y. Zou, Venkatesh Saligrama, and Adam Kalai. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. *CoRR*, abs/1607.06520, 2016.

[7] Nal Kalchbrenner and Phil Blunsom. Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1700–1709, Seattle, Washington, USA, October 2013. Association for Computational Linguistics.

[8] Hany Hassan, Anthony Aue, Chang Chen, Vishal Chowdhary, Jonathan Clark, Christian Federmann, Xuedong Huang, Marcin Junczys-Dowmunt, William Lewis, Mu Li, Shujie Liu, Tie-Yan Liu, Renqian Luo, Arul Menezes, Tao Qin, Frank Seide, Xu Tan, Fei Tian, Lijun Wu, Shuangzhi Wu, Yingce Xia, Dongdong Zhang, Zhirui Zhang, and Ming Zhou. Achieving human parity on automatic chinese to english news translation. *CoRR*, abs/1803.05567, 2018.

[9] David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, jan 2016.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *CoRR*, abs/1502.01852, 2015.

[11] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. *CoRR*, abs/1409.3215, 2014.

[12] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proc. of the Conference on EMNLP*, pages 1724–1734, 2014.

[13] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[14] Henry J Kelley. Gradient theory of optimal flight paths. *Ars Journal*, 30(10):947–954, 1960.

[15] Jianpeng Cheng, Li Dong, and Mirella Lapata. Long short-term memory-networks for machine reading. *CoRR*, abs/1601.06733, 2016.

[16] Zhouhan Lin, Minwei Feng, Cícero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. A structured self-attentive sentence embedding. *CoRR*, abs/1703.03130, 2017.

[17] Ankur P. Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. A decomposable attention model for natural language inference. *CoRR*, abs/1606.01933, 2016.

[18] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[20] Alec Radford. Improving language understanding by generative pre-training. 2018.

[21] Philipp Koehn and Rebecca Knowles. Six challenges for neural machine translation. *CoRR*, abs/1706.03872, 2017.

[22] Mingsheng Long, Jianmin Wang, and Michael I. Jordan. Unsupervised domain adaptation with residual transfer networks. *CoRR*, abs/1602.04433, 2016.

[23] Chenhui Chu and Rui Wang. A survey of domain adaptation for neural machine translation. *CoRR*, abs/1806.00258, 2018.

[24] Rico Sennrich, Barry Haddow, and Alexandra Birch. Improving neural machine translation models with monolingual data. *CoRR*, abs/1511.06709, 2015.

[25] Hoang Cuong and Khalil Sima'an. Latent domain translation models in mix-of-domains haystack. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1928–1939, 2014.

[26] Nadir Durrani Hassan Sajjad Shafiq Joty and Ahmed Abdelali Stephan Vogel. Using joint models for domain adaptation in statistical machine translation. *Proceedings of MT Summit XV*, page 117, 2015.

[27] Boxing Chen, Roland Kuhn, George Foster, Colin Cherry, and Fei Huang. Bilingual methods for adaptive training data selection for machine translation. In *Proc. of AMTA*, pages 93–103, 2016.

[28] Chenhui Chu. Integrated parallel data extraction from comparable corpora for statistical machine translation. 2015.

[29] Rui Wang, Hai Zhao, Bao-Liang Lu, Masao Utiyama, and Eiichiro Sumita. Neural network based bilingual language model growing for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 189–195, Doha, Qatar, October 2014. Association for Computational Linguistics.

[30] Rui Wang, Hai Zhao, Bao-Liang Lu, Masao Utiyama, and Eiichiro Sumita. Connecting phrase based statistical machine translation adaptation. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3135–3145, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee.

[31] Benjamin Marie and Atsushi Fujita. Efficient extraction of pseudo-parallel sentences from raw monolingual data using word embeddings. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 392–398, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[32] Rico Sennrich, Holger Schwenk, and Walid Aransa. A multi-domain translation model framework for statistical machine translation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 832–840, Sofia, Bulgaria, August 2013. Association for Computational Linguistics.

[33] Kenji Imamura and Eiichiro Sumita. Multi-domain adaptation for statistical machine translation based on feature augmentation. *Journal of Natural Language Processing*, 24:597–618, 09 2017.

[34] Anthony Rousseau, Fethi Bougares, Paul Deléglise, Holger Schwenk, and Yannick Estève. Lium's systems for the iwslt 2011 speech translation tasks. In *International Workshop on Spoken Language Translation (IWSLT) 2011*, 2011.

[35] Tobias Domhan and Felix Hieber. Using target-side monolingual data for neural machine translation through multi-task learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1500–1505, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[36] Jiajun Zhang and Chengqing Zong. Exploiting source-side monolingual data in neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545, Austin, Texas, November 2016. Association for Computational Linguistics.

[37] Yong Cheng, Wei Xu, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. Semi-supervised learning for neural machine translation. *CoRR*, abs/1606.04596, 2016.

[38] Chenhui Chu, Raj Dabre, and Sadao Kurohashi. An empirical comparison of domain adaptation methods for neural machine translation. In *Proceedings of the 55th Annual*

Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 385–391, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[39] Rico Sennrich, Barry Haddow, and Alexandra Birch. Controlling politeness in neural machine translation via side constraints. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 35–40, San Diego, California, June 2016. Association for Computational Linguistics.

[40] Hassan Sajjad, Nadir Durrani, Fahim Dalvi, Yonatan Belinkov, and Stephan Vogel. Neural machine translation training in a multi-domain scenario. *CoRR*, abs/1708.08712, 2017.

[41] Boxing Chen, Colin Cherry, George Foster, and Samuel Larkin. Cost weighting for neural machine translation domain adaptation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 40–46, Vancouver, August 2017. Association for Computational Linguistics.

[42] Praveen Dakwale. Fine-tuning for neural machine translation with limited degradation across in-and out-of-domain data. 2017.

[43] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015.

[44] Çaglar Gülçehre, Orhan Firat, Kelvin Xu, Kyunghyun Cho, Loïc Barrault, Huei-Chi Lin, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. On using monolingual corpora in neural machine translation. *CoRR*, abs/1503.03535, 2015.

[45] Denny Britz, Quoc Le, and Reid Pryzant. Effective domain mixing for neural machine translation. In *Proceedings of the Second Conference on Machine Translation*, pages 118–126, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[46] Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. Gender bias in coreference resolution: Evaluation and debiasing methods. *arXiv preprint arXiv:1804.06876*, 2018.

[47] Ran Zmigrod, Sabrina J. Mielke, Hanna Wallach, and Ryan Cotterell. Counterfactual data augmentation for mitigating gender stereotypes in languages with rich morphology. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1651–1661, Florence, Italy, July 2019. Association for Computational Linguistics.

[48] Joel Escudé Font and Marta R. Costa-jussà. Equalizing gender bias in neural machine translation with word embeddings techniques. In *Proceedings of the First ACL Workshop on Gender Bias in Natural Language Processing*, pages 147–154, Florence, Italy, August 2019.

[49] Danielle Saunders and Bill Byrne. Reducing gender bias in neural machine translation as a domain adaptation problem. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7724–7736, Online, July 2020. Association for Computational Linguistics.

[50] James Kirkpatrick, Razvan Pascanu, Neil C. Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *CoRR*, abs/1612.00796, 2016.

[51] Felix Stahlberg, Eva Hasler, Aurelien Waite, and Bill Byrne. Syntactically guided neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 299–305, Berlin, Germany, August 2016. Association for Computational Linguistics.

[52] Philipp Koehn. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86. Citeseer, 2005.

[53] Giusepppe Attardi. Wikiextractor. `https://github.com/attardi/wikiextractor`, 2015.

[54] Mikel Artetxe and Holger Schwenk. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *CoRR*, abs/1812.10464, 2018.

[55] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff 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., 2013.

[56] Aylin Caliskan, Joanna J. Bryson, and Arvind Narayanan. Semantics derived automatically from language corpora necessarily contain human biases. *Science*, 356:183–186, 2017.

[57] Jieyu Zhao, Yichao Zhou, Zeyu Li, Wei Wang, and Kai-Wei Chang. Learning gender-neutral word embeddings. *CoRR*, abs/1809.01496, 2018.

[58] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.

[59] Chenhui Chu and Raj Dabre. Multilingual multi-domain adaptation approaches for neural machine translation. *CoRR*, abs/1906.07978, 2019.

[60] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alex Constantin, and Evan Herbst. Moses: Open source toolkit for statistical machine translation. 06 2007.

[61] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics.

Appendices

Transformer Parameters The full list of parameters that were used in the training of the baseline model. In the finetuning step, the dropout was changed to 0.3 in order to have a better generalization and avoid adapting too much to the gender-balanced corpus.

Parameter	Value
Adam Betas	$(0.9, 0.98)$
Adam Epsilon	10^{-8}
Attention Dropout	0.1
Clip Norm	0.0
Criterion	Label Smoothed Cross Entropy
Label Smoothing	0.1
Encoder	Input Dim=512, Output Dim=512, Layers=6, Attention Heads=8, Embed Dim=512, FFN Embed Dim=2048
Decoder	Input Dim=512, Output Dim=512, Layers=6, Attention Heads=8, Embed Dim=512, FFN Embed Dim=2048
Warmup Policy	$(10^{-7}, 4 \times 10^{3})$
Dropout	0.1
Learning Rate	0.001
Learning Rate Scheduler	Inverse Square Root
Max Source Positions	1024
Max Target Positions	1024
Max Tokens	3584
Min Learning Rate	1^{-9}
Momentum	0.99
Batch Size	16

Table 8: Parameters used for training. These were the parameters that gave the best performance on the baseline model.