Generating Gender Alternatives in Machine Translation

Sarthak Garg[§], Mozhdeh Gheini[*¶], Clara Emmanuel[§], Tatiana Likhomanenko[§], Qin Gao[†§], Matthias Paulik[†§]

[§]Apple Inc.
{sarthak_garg,clara_e,antares,qingao,mpaulik}@apple.com
[¶]Information Sciences Institute, University of Southern California
gheini@isi.edu

Abstract

Machine translation (MT) systems often translate terms with ambiguous gender (e.g., English term "the nurse") into the gendered form that is most prevalent in the systems' training data (e.g., "enfermera", the Spanish term for a female nurse). This often reflects and perpetuates harmful stereotypes present in society. With MT user interfaces in mind that allow for resolving gender ambiguity in a frictionless manner, we study the problem of generating *all* grammatically correct gendered translation alternatives. We open source train and test datasets for five language pairs and establish benchmarks for this task. Our key technical contribution is a novel semi-supervised solution for generating alternatives that integrates seamlessly with standard MT models and maintains high performance without requiring additional components or increasing inference overhead.

1 Introduction and Related Work

Gender[1] biases present in train data are known to bleed into natural language processing (NLP) systems, resulting in dissemination and potential amplification of those biases (Sun et al., 2019). Such biases are often also the root cause of errors. A machine translation (MT) system might, for example, translate *doctor* to the Spanish term *médico* (masculine) instead of *médica* (feminine), given the input "The doctor asked the nurse to help her in the procedure" (Stanovsky et al., 2019). To avoid prescribing wrong gender assignment, MT systems need to disambiguate gender through context. When the correct gender cannot be determined through context, providing multiple translation alternatives that cover all valid gender choices is a reasonable approach.

Numerous prior works have focused on producing correctly gendered translations given contextual gender "hints", such as "to help her" in the example above (Stanovsky et al., 2019; Saunders and Byrne, 2020; Stafanovičs et al., 2020; Costa-jussà et al., 2022; Saunders et al., 2022; Renduchintala et al., 2021; Bentivogli et al., 2020; Currey et al., 2022). In contrast, the problem of generating all valid and grammatically correct gendered translations has seen far less attention (Kuczmarski and Johnson, 2018; Johnson, 2020; Sánchez et al., 2023).

Consider the example: "The secretary was angry with the boss." The gender of both *secretary* and *boss* remain ambiguous in the absence of additional context: both entities can take either gender. However, and to the best of our knowledge, all existing approaches (Kuczmarski and Johnson, 2018; Johnson, 2020; Sánchez et al., 2023; Rarrick et al., 2023) for producing different gendered translations operate on "sentence-level", instead of on "entity-level": they only allow two sentence-level alternatives to surface, in which both *secretary* and *boss* are either masculine or feminine:

- secretary, boss: El secretario estaba enojado con el jefe.[2]
- secretary, boss: La secretaria estaba enojada con la jefa.

In this work, we introduce a novel approach that operates on entity-level, i.e., it generates four alternatives corresponding to all grammatically valid combinations of gender choices for both entities:

- secretary, boss: El secretario estaba enojado con el jefe.
- secretary, boss: El secretario estaba enojado con la jefa.
- secretary, boss: La secretaria estaba enojada con el jefe.
- secretary, boss: La secretaria estaba enojada con la jefa.

When integrated with a proper user interface, our approach provides users with the freedom to choose gender for each entity. We posit that any such system should meet the following practical quality criteria, making the problem challenging:

[*] Work done during an internship at Apple.

[†] Equal senior contribution.

[1] "gender" in this work refers to binary grammatical gender, and not social gender (male, female, nonbinary). Please refer to §Limitations for a detailed discussion.

[2] Gendered translations in Spanish. Brown and teal represent masculine and feminine genders respectively.

Proceedings of the 5th Workshop on Gender Bias in Natural Language Processing (GeBNLP), pages 237–254
August 16, 2024 ©2024 Association for Computational Linguistics

- Alternatives should not be produced when the gender can be inferred from the sentence context, e.g., "She is a boss" should only produce the feminine translation "Ella es una jefa".

- All alternatives should maintain grammatical gender agreement. Phrases like "El secretaria" or "secretaria estaba enojado" should not be produced as they break gender agreement by using different gendered forms for the same entity.

- Alternatives should differ *only* in gender inflections and not general wording, formality, etc., as any such differences can potentially encode bias.

This paper presents several key contributions towards studying the task of generating entity-level alternatives, meeting the above quality criteria:

- Producing entity-level alternatives for n gender-ambiguous entities requires generating 2^n different translations. We propose an efficient approach that reduces the problem to generating a *single* structured translation where "gender-sensitive phrases" are grouped together and aligned to corresponding ambiguous entities.

- We open source train datasets [3] for this task for 5 language pairs and establish supervised baselines. We extend an existing test set for this task: GATE (Rarrick et al., 2023) from 3 to 6 language pairs and open source the extended set.

- We develop a semi-supervised approach that leverages pre-trained MT models or large language models (LLMs) for data augmentation. Models trained on augmented data outperform the supervised baselines and can also generalize to language pairs not covered in the train sets.

2 Entity-Level Gender Alternatives

Our key insight for efficiently generating entity-level gender alternatives is to reduce the problem to generating a single translation with embedded *gender structures* and their *gender alignments*.

Consider our previous example: "The secretary was angry with the boss." We want to generate the following entity-level alternatives:

- secretary, boss: El secretario estaba enojado con el jefe.
- secretary, boss: El secretario estaba enojado con la jefa.
- secretary, boss: La secretaria estaba enojada con el jefe.
- secretary, boss: La secretaria estaba enojada con la jefa.

[3] https://github.com/apple/ml-gendered-translation

Since we constraint the alternatives to only differ in gender inflections, we can instead produce a single translation with gender-sensitive phrases grouped together as gender structures, shown in $\binom{}{}$:

$$\binom{\text{El secretario}}{\text{La secretaria}} \text{ estaba } \binom{\text{enojado}}{\text{enojada}} \text{ con } \binom{\text{el jefe}}{\text{la jefa}}$$

All alternatives can be derived from this single translation by choosing either the masculine or feminine form in each gender structure. However, doing this naively can give us invalid alternatives that break gender agreement, for example:

El secretario estaba enojada con el jefe

$\binom{\text{El secretario}}{\text{La secretaria}}$ and $\binom{\text{enojado}}{\text{enojada}}$ correspond to the same entity *secretary* and cannot have different gender choices. By having gender alignments between each gender structure in the translation and its corresponding gender-ambiguous entity in the source, we can deduce which gender structures are linked together and need to be consistent with each other.

Let $x = x_1 \dots x_n$ be the source sentence containing n tokens and let $G_a \subseteq \{1 \dots n\}$ represent the set of indices of gender-ambiguous entities in x. We aim to produce a translation y_S:

$$y_S = y_1 \dots \binom{M_1}{F_1} \dots \binom{M_k}{F_k} \dots y_m, \qquad (1)$$

containing a set of gender structures $S = \{S_1 \dots S_k\}$ where $S_i := \binom{M_i}{F_i}$ is the i^{th} gender structure. Translation y_S is a sequence of two types of elements: $\{y_1 \dots y_m\} = y_S \setminus S$ are regular tokens that do not change based on the gender of any entity in G_a and M_*/F_* are the masculine and feminine inflected forms of the phrases that do change based on the gender of an entity in G_a. Gender alignments can then be formally defined as a one-to-many mapping from G_a to S. An ambiguous entity is aligned to a gender structure $\binom{\text{M}}{\text{F}}$ iff the correct inflection form (M or F) in the translation depends on the gender of the entity. In our example, *secretary* is aligned to $\binom{\text{El secretario}}{\text{La secretaria}}$, $\binom{\text{enojado}}{\text{enojada}}$, and *boss* is aligned to $\binom{\text{el jefe}}{\text{la jefa}}$. Given the translation with gender structures y_S and gender alignments, alternatives corresponding to any combination of gender assignments of ambiguous entities can be easily derived as follows: for all ambiguous entities with male gender assignment, choose the male form for their aligned gender structures. Similarly, for all entities with female assignments, choose the female form for their aligned gender structures.

Source annotations	**Target annotations**	**Alignment annotations**
The **lawyer** fought to keep his **child**, who is a gangster, safe from the **judge**. **lawyer** → Masculine **child** → Gender-Ambiguous **judge** → Gender-Ambiguous	El abogado luchó para mantener a su $\binom{\text{hijo}}{\text{hija}}$, que es $\binom{\text{un}}{\text{una}}$ gángster, a salvo $\binom{\text{del juez}}{\text{de la jueza}}$.	**child** → $\binom{\text{hijo}}{\text{hija}}$, $\binom{\text{un}}{\text{una}}$ **judge** → $\binom{\text{del juez}}{\text{de la jueza}}$

Table 1: English–Spanish annotation example. *lawyer*, *child* and *judge* are the annotated entities. *child* and *gangster* refer to the same entity and *child* is selected as the head-word. *lawyer* is marked as masculine because of the co-referring pronoun *his* and is translated to the masculine form: *El abogado*. *child* and *judge* are gender-ambiguous leading to gender structures in the translation (middle column) and gender alignments (rightmost column).

3 Datasets

To build and evaluate systems producing alternatives, we prepare train and test sets containing gender structures and gender alignment annotations.

3.1 Test data

We evaluate our models on a combination of two existing test sets that test complementary aspects:

- GATE (Rarrick et al., 2023) has source sentences with at least 1 and at most 3 gender-ambiguous entities with their entity-level alternatives satisfying our quality criteria. It evaluates the system on cases where alternatives *should* be produced.
- MT-GenEval (Currey et al., 2022) contains sentences with entities whose gender can be inferred from the sentence context and are not ambiguous. This set is helpful for evaluating cases where alternatives *should not* be produced.

These two test sets have different annotation formats and guidelines. In order to unify them, we ask annotators to review and post-edit existing annotations using the following guidelines:

1. **Marking gendered words**: First, all words in the source referring to entities (people/animals) that can have masculine or feminine grammatical genders are marked.
2. **Gender ambiguity annotation**: Next, if multiple words refer to the same entity, a head word is selected among them. We guided the annotators to pick the one that acts the most like the subject as the head word. For each head word, if its gender can be inferred from the grammatical context, such as co-referring male/female pronouns, it is marked as such. If no gender can be inferred, the gender is marked as ambiguous. We only rely on grammatical sentence context and not on external knowledge/common gender associations of names/proper nouns. Appendix B discusses how our annotation guidelines handle the problem of masculine generics (Piergentili et al., 2023a), where masculine nouns/pronouns can be used to refer to ambiguous or collective entities.
3. **Gender aware translation**: Finally, we ask the annotators to translate the source sentence. Entities without any ambiguity must be translated into the correct gender. If the translation depends on the gender of the ambiguous entities in the source, gender structures and gender alignments are annotated.

Table 1 explains the process with the help of an example annotation. We prepare this unified test set for 6 language pairs: English to German, French, Spanish, Portuguese, Russian, and Italian.[4]

3.2 Train data

We open source train data containing samples in the same format as the test set to ensure reproducibility and to encourage development of supervised/semi-supervised systems for producing alternatives. In contrast to the test sets, which are created via human annotation, we rely on an automatic data augmentation approach (see Appendix C for details) to create train data at scale. The source sentences for the train sets are sampled from Europarl (Koehn, 2005), WikiTitles (Tiedemann, 2012), and WikiMatrix (Schwenk et al., 2021) corpora. The train data are partitioned into two different sets:

- **G-Tag** contains source sentences with head words for all entities with their gender ambiguity label: Masc., Fem. or Ambiguous.
- **G-Trans** contains gender-ambiguous entities in the source sentences, gender structures in the translations and gender alignments.

To the best of our knowledge, this is the first large-scale corpus that contains gender ambiguities and how they effect gendered forms in the translation. We release these sets for 5 language pairs: English to German, French, Spanish, Portuguese, and Russian. G-Tag contains $\sim$ 12k sentences and

[4]We extend the original GATE corpus, which only includes English to Spanish, French, and Italian.

G-Trans contains $\sim 50k$ sentence pairs per language pair. Detailed statistics of the train and test sets can be found in Appendix A.

4 Training MT Models to Generate Gender Structures and Alignments

We first present how to train MT models that produce gender structures and alignments, assuming parallel data enriched with gender structures and alignments (for example, G-Trans) is available. We then describe a novel data augmentation pipeline that can enrich any regular parallel corpora with gender structures and alignments.

Given a source sentence $x = \{x_1 \dots x_n\}$, translation y_S containing gender structures, and gender alignments A, we want to train the MT model to generate $y_S, A|x$. Let's assume that y_S contains k gender structures and $A = \{a_1 \dots a_k\}$ where a_i represents the source token aligned to the i^{th} gender structure. We serialize each gender structure in y_S into a sequence of tokens as follows:

$$\left(\begin{smallmatrix}\text{M}\\\text{F}\end{smallmatrix}\right) \rightarrow \texttt{BEG M MID F END}$$

where BEG, MID, and END are special tokens. The model is then trained to produce gender structures in the form of this sequence.

Garg et al. (2019) introduced a technique to train MT models to jointly generate translations and word-alignments. We use their approach to learn generation of gender alignments. Let $m_1 \dots m_k$ denote the positions of the MID tokens of the gender structures. A specific cross-attention head is chosen and supervised to learn gender alignments. Let n and m denote the lengths of the source and the serialized target respectively and let $P_{m \times n}$ denote the attention probability distribution computed by the selected head. We train the model with regular cross entropy and an additional *alignment loss*:

$$L = L_{\text{cross-ent}} - \frac{\lambda}{k} \sum_{i=1}^{k} log(P_{m_i, a_i})$$

where λ is a scaling factor. This added loss term encourages the attention head to place more probability mass on the aligned source token when generating the MID token belonging to that token's gender structure. During inference, the gender alignment for the i^{th} gender structure can be computed as:

$$a_i = \underset{s \in \{x_1 \dots x_n\}}{\text{argmax}} \; P_{m_i, s}$$

This model can generate gender structures and alignments without any additional inference overhead. Then, using the procedure described in section 2, all entity-level alternatives can be easily derived from the model outputs.

5 Data Augmentation Pipeline

G-Trans dataset provides supervised data to train MT models in the above manner. However, this dataset is small ($50k$ examples per language pair) and has a restrictive domain, limiting the quality of the trained models. We propose a *data augmentation* pipeline that can take any regular parallel corpora (containing high quality but potentially biased translations) and augment the translations with gender structures and alignments whenever there are ambiguities in the source.

Algorithm 1 Data Augmentation Overview

Input: $x = \{x_1 \dots x_n\}$ (source sentence) and $y_B = \{y_1 \dots y_m\}$ (reference translation without gender structures, potentially biased)

▷ **Step 1:** Detect set of gender-ambiguous entities G_a in the source sentence: $G_a \subseteq \{1 \dots n\}$
$G_a \leftarrow \text{GenderAmbiguousEntities}(x)$
if $G_a = \phi$ **then**
  Output: x, y_B, ϕ
end if

▷ **Step 2:** Transform y_B into an all-masculine y_M and all-feminine y_F translations
$y_M \leftarrow \text{argmax } p(y|x, y_B, \text{gender}(x_i) = \text{male } \forall i \in G_a)$
$y_F \leftarrow \text{argmax } p(y|x, y_B, \text{gender}(x_i) = \text{female } \forall i \in G_a)$

▷ **Step 3:** Combine y_M and y_F into a single translation y_S containing gender structures
Let $y_M = y_1 \dots M_1 \dots y_j \dots M_k \dots y_m$ and
Let $y_F = y_1 \dots F_1 \dots y_j \dots F_k \dots y_m$
where y_* are the common tokens between y_M and y_F and $\{(M_i, F_i) \mid i \in 1 \dots k\}$ be the k differing phrases.
$y_S \leftarrow \text{group}(y_M, y_F) = y_1 \dots \left(\begin{smallmatrix}M_1\\F_1\end{smallmatrix}\right) \dots \left(\begin{smallmatrix}M_k\\F_k\end{smallmatrix}\right) \dots y_m$

▷ **Step 4:** Align each gender structure $S_i := \left(\begin{smallmatrix}M_i\\F_i\end{smallmatrix}\right)$ to its corresponding ambiguous entity in G_a
$A \leftarrow \text{ComputeGenderAlignments}(x, y_S)$
Output: x, y_s, A

Algorithm 1 gives an overview of the main components of the pipeline, which we describe in detail in the following subsections. It consists of first detecting gender-ambiguous entities in the source sentence (§5.1), followed by transforming the reference translation into all-masculine/all-feminine translations (§5.2, §5.3), condensing those into single translation with gender structures, and finally aligning the gender structures (§5.4).

	Source	Target
G-Trans dataset	The doctor was angry with the patient doctor → Gender-Ambiguous patient → Gender-Ambiguous	(El doctor / La doctora) estaba (enojado / enojada) con (el / la) paciente
Fine-tuning bi-text	The doctor<M> was angry with the patient<M>	*El doctor* estaba *enojado* con *el* paciente
	The doctor<F> was angry with the patient<F>	*La doctora* estaba *enojada* con *la* paciente
	The doctor<M> was angry with the patient<F>	*El doctor* estaba *enojado* con *la* paciente
	The doctor<F> was angry with the patient<M>	*La doctora* estaba *enojada* con *el* paciente

Table 2: Extracting bi-text for fine-tuning from the G-Trans dataset. Each gender-ambiguous token is suffixed with a gender assignment tag: <M>/<F>. With the help of alignments (shown via color coding), the correct gender inflection is selected in the translation. n ambiguous entities can result in 2^n different assignments, but we only keep "all-masculine", "all-feminine", and a maximum of 3 other randomly sampled assignments.

5.1 Detecting gender-ambiguous entities

Traditionally, rule-based methods, which rely on dependency parsing and co-reference resolution, are used to detect gender-ambiguous entities in the source sentence (Rarrick et al., 2023; Habash et al., 2019). In contrast, we adopt a data-driven approach. G-Tag dataset contains English source sentences annotated with head-words, which refer to entities with their gender label derived from the grammatical sentence context: ambiguous, masculine, feminine. Following Alhafni et al. (2022), we fine-tune a (BERT-style) pre-trained language model (PLM) using this dataset to tag each source token with one of the four labels: ambiguous, masculine, feminine, or not a headword.

5.2 Generating all-masculine/feminine translations using fine-tuned MT models

If ambiguous entities are detected in the source sentence, then the next step is to transform the high-quality but potentially biased reference translation y_B to all-masculine y_M and all-feminine y_F translations. y_M and y_F are equivalent to sentence-level alternatives corresponding to masculine and feminine assignments for all ambiguous entities, respectively. We explore two methods for this task: fine-tuning pre-trained MT models (this subsection) and using LLMs (subsection 5.3).

We fine-tune a pre-trained MT model M on a bi-text extracted from the G-Trans dataset. The source sentences of this bi-text contain ambiguous entities tagged as masculine or feminine using <M>/<F> tags, and the target translation has correct gender inflections given the gender tags. Table 2 explains this extraction process in detail using an example.

The fine-tuned model $M_{\text{fine-tuned}}$ learns to generate translations with gender inflections in agreement with the gender assignments (<M>/<F>) in the source. We use Saunders and Byrne (2020)'s lattice rescoring approach to generate y_M and y_F. Let x_M and x_F denote source sentences in which all ambiguous entities (G_a) have been tagged using <M> and <F> tags, respectively. Let $I(y_B)$ represent the search space consisting only of all possible gender inflection variants of y_B. $M_{\text{fine-tuned}}$ is used to decode y_M and y_F over the constrained search space $I(y_B)$:

$$y_M = \underset{y \in I(y_B)}{\operatorname{argmax}}\, p_{M_{\text{fine-tuned}}}(y|x_M)$$

$$y_F = \underset{y \in I(y_B)}{\operatorname{argmax}}\, p_{M_{\text{fine-tuned}}}(y|x_F)$$

This can be done efficiently using constrained beam search. This procedure guarantees that y_M, y_F, and y_B differ only in terms of gender inflections, and therefore, y_M and y_F possess the same general translation quality as the reference translation y_B.

In-Context Example(s)

English input: That is not what the taxpayers and consumers of Europe want.
Base Spanish translation: No es eso lo que quieren los contribuyentes y los consumidores de Europa.
Edited Spanish translation (feminine taxpayers, feminine consumers): No es eso lo que quieren las contribuyentes y las consumidoras de Europa.
Edited Spanish translation (masculine taxpayers, masculine consumers): No es eso lo que quieren los contribuyentes y los consumidores de Europa.
…

Test Input

English input: $\{x\}$
Base Spanish translation: $\{y_B\}$
Edited Spanish translation ($\{$feminine i $\forall i \in G_a\}$):

LLM Output

$\{y_F\}$
Edited Spanish translation ($\{$masculine i $\forall i \in G_a\}$): $\{y_M\}$

Figure 1: Prompting LLMs using in-context examples to edit the reference translation y_B into all-masculine and all-feminine gender assignments. Multiple in-context examples are used but we illustrate only one here for brevity.

5.3 Generating all-masculine/feminine translations using LLMs

LLMs' ability to learn using in-context examples (Brown et al., 2020) provides us with an alternative approach for generating y_M and y_F. We can provide selected instances from G-Trans as in-context examples in the prompt to the LLM and have it generate output for a test instance (Sánchez et al., 2023). Inspired by re-writing literature (Vanmassenhove et al., 2021; Sun et al., 2021), we design a prompt that treats the LLM as an editor: it edits/re-writes the given translation y_B to match the provided gender assignments (all-masculine and all-feminine) in the prompt (See Figure 1 for an example).

5.4 Aligning gender structures

y_M and y_F are combined together in Step 3 as described in Algorithm 1 to produce a single translation y_S containing gender structures. The final step is to align each gender structure in y_S to an ambiguous entity in the source. We model this as a tagging task and fine-tune a PLM using alignment annotations in the G-Trans dataset.

Algorithm 2 Alignment Algorithm

Input: $x = \{x_1 \dots x_n\}$ (source sentence) and y_S (translation with k gender structures)

Let $y_S = y_1 \dots \left(\begin{smallmatrix} M_1 \\ F_1 \end{smallmatrix}\right) \dots \left(\begin{smallmatrix} M_k \\ F_k \end{smallmatrix}\right) \dots y_m$

for i^{th} gender structure $S_i := \left(\begin{smallmatrix} M_i \\ F_i \end{smallmatrix}\right)$ **do**

  Let | be a special marker token

  $y_A \leftarrow y_1 \dots M_1 \dots |M_i| \dots M_k \dots y_m$

  $a_i \leftarrow \text{PLM}(x; y_A)$ ▷ ; denotes concatenation

end for

Output: $A = \{a_i, \forall i \in 1 \dots k\}$

Each gender structure is aligned one-by-one as described in Algorithm 2. To align the i^{th} gender structure S_i, we take y_M and enclose the phrase corresponding to S_i by a special token | to get y_A. Then x and y_A are concatenated together and fed to the PLM, which is fine-tuned to tag all the tokens in x as aligned/not-aligned to S_i (See Figure 5 in the Appendix for an example). The gold aligned/not-aligned labels for fine-tuning are extracted from the G-Trans dataset.

6 Evaluation Metrics

We evaluate our systems' performance using the following metrics:

- **Alternatives metrics**: These metrics compute the overlap between the set of sentences that have alternatives in the test set and the set of sentences for which the system produces alternatives. This overlap is measured using precision and recall and gives a sense of how often the system produces alternatives and whether it produces them only when needed.
- **Structure metrics**: These metrics are computed over the set of sentences for which both the test set and system output contain alternatives. They measure the quality of the generated alternatives by computing the overlap between the gender structures in the reference alternatives and the generated alternatives. The overlap is measured using precision and recall.
- **Alignment accuracy**: This is measured as the % of gender structures that are aligned to the correct source entity and reflects the quality of gender agreement in the generated alternatives.
- **δ-BLEU**: Lastly, following Currey et al. (2022), to measure the degree of bias towards a gender, we compute *δ-BLEU* as follows: We separate the masculine and feminine forms in gender structures (if any) for the reference and the system output, compute masculine and feminine BLEU scores (using `sacrebleu` (Post, 2018)), and measure the absolute difference between the two:

$$\delta\text{-BLEU} = |\text{BLEU}(\hat{y}_m, y_m) - \text{BLEU}(\hat{y}_f, y_f)|$$

Higher δ-BLEU indicates more bias. Mathematical definitions of alternatives and structure metrics can be found in Appendix K.

7 Experiments and Results

We will first describe the experimental details and results of our data augmentation pipeline in 7.1 and 7.2. We then present the training details of the MT model generating alternatives end-to-end and how it benefits from data augmentation in 7.3 and 7.4.

7.1 Data augmentation pipeline details

The data augmentation pipeline consists of three components: detecting gender-ambiguous entities, generating all-masculine/feminine translations and aligning gender structures.

We build the ambiguous entity detector (§5.1) by fine-tuning `xlm-roberta-large` (Conneau et al., 2020) using `transformers` (Wolf et al., 2020). We use the combined G-Tag dataset across all 5 language pairs for fine-tuning.

To generate all-masculine/feminine translations, we explore two approaches: fine-tuning pre-trained MT models (§5.2), and using LLMs (§5.3). For the first approach, we fine-tune the `M2M 1.2B` (Fan et al., 2021) model using `fairseq` (Ott et al., 2019). The model is fine-tuned jointly on bi-text

Language Pair	Model	Alternatives Metrics ↑ Precision%	Recall%	δ-BLEU ↓	Structure Metrics ↑ Precision%	Recall%	Alignment ↑ Accuracy%
En–De	Fine-tuned M2M	**94**	89.7	4.7	87.8	91	93.7
	GPT	91.1	**92.7**	**2.8**	**89.8**	**94**	
En–Es	Fine-tuned M2M	**95.7**	91.6	3.3	**88.1**	**93**	91.5
	GPT	91.5	**93.7**	**2.7**	84.7	92.7	
En–Fr	Fine-tuned M2M	**93.8**	**92.5**	3.6	**88.1**	92.9	92.9
	GPT	89.4	91	**2.8**	85.8	**94.8**	
En–Pt	Fine-tuned M2M	**94.8**	**94.3**	**3.5**	88.3	92.4	93.6
	GPT	93.8	83.5	5.5	**89.6**	**95.2**	
En–Ru	Fine-tuned M2M	**89.4**	**89.3**	**5.7**	**87**	**87.7**	93.2
	GPT	83.5	58.2	10.6	83.1	85	
En–It	Fine-tuned M2M	95.4	87.9	8.2	79.4	75.3	94.1

Table 3: Data augmentation pipeline results. ↑ indicates higher-the-better and ↓ lower-the-better metrics.

extracted from the G-Trans dataset (as described in Table 2) for all 5 language pairs. The list of gender inflections used for lattice rescoring is collected from Wiktionary (Ylonen, 2022) and inflections present in the G-Trans train and test sets.

For the second approach, we use `gpt-3.5-turbo` as our LLM and follow the prompt design described in subsection 5.3 with 6 in-context examples. We provide additional ablation studies on the number of in-context examples, different prompt designs, and choice of LLM (`gpt` vs. `OpenLlama-v2-7B` (Geng and Liu, 2023)) in Appendix F. We find that using more in-context examples helps, but gains are minimal for > 6. Since LLM decoding does not use lattice rescoring, it is possible that the generated all-masculine/feminine translations differ in more than just gender inflections. To avoid this, we explicitly check the differences and don't generate gender structures if the differences don't match any entry in the list of gender inflections.

Lastly, to align gender structures we fine-tune `xlm-roberta-large` on source, targets, and gender alignments extracted from the G-Trans dataset jointly for all 5 language pairs. The hyper-parameters for fine-tuning XLM and M2M models are decided based on validation performance on a held-out portion of the train sets and can be found in Appendices D, E and G.

7.2 Data augmentation pipeline results

The data augmentation pipeline takes source sentences and their reference translations (without gender structures, potentially biased) as inputs. For evaluating the data augmentation pipeline, we feed in the source sentences and their all-masculine reference translations from the test set as inputs. The pipeline returns these translations augmented with gender structures and alignments. We can then compute the evaluation metrics described in section 6 on the generated gender structures and alignments. Table 3 summarizes the results.

Both M2M and GPT perform mostly on par with the exception of English-Russian, where GPT achieves much lower alternatives recall (58.7 compared to 89.3). The quality of generated gender structures is better for GPT on English-German and English-Portuguese and better for M2M on English-Spanish and English-Russian, as can be seen from the structure metrics. Note that we don't have any G-Trans data for English-Italian, so the results of the M2M model and the alignment accuracy on English-Italian are purely due to zero-shot generalization of M2M and XLM models (Johnson et al., 2017). Overall, the zero-shot results are comparable to others in terms of alternatives metrics and alignment accuracy but fall behind on structure metrics. The alignment model performs well obtaining $\geq 91\%$ accuracy on all language pairs.

δ-BLEU depends on both alternatives and structure metrics and can be used as a single metric to compare systems' performance. Overall, GPT wins in terms of not relying on any fine-tuning dataset and better performance on English to German, Spanish, and French. Fine-tuning M2M wins in terms of achieving better results on English to Portuguese and Russian and being much more efficient in terms of parameters and inference cost (`M2M 1.2B` can be fit on a single A100 GPU).

Finally, Table 5 compares the performance of our data augmentation pipeline using M2M against GATE's sentence-level gender re-writer on their setup. We use our pipeline to re-write an all-masculine reference into an all-feminine form (M→F) and vice-versa (F→M). More details about

Language Pair	Model	Alternatives Metrics ↑ P%	Alternatives Metrics ↑ R%	δ-BLEU ↓	Structure Metrics↑ P%	Structure Metrics↑ R%	Alignment Accuracy ↑ %	FLoRes BLEU ↑
En–De	Vanilla	-	-	8.6	-	-	-	31.6
	Supervised	74.4	71.5	2.4	**55.2**	**57.5**	89.1	**31.9**
	w/ Augmented Data	**86.7**	**87.5**	**0.8**	48.2	55.6	**94.2**	31.6
En–Es	Vanilla	-	-	10.4	-	-	-	**26**
	Supervised	78.9	77.3	2.8	60.5	60.6	85.2	25.9
	w/ Augmented Data	**94.3**	**92**	**1**	**62.4**	**66.4**	**92.5**	**26**
En–Fr	Vanilla	-	-	8.1	-	-	-	**46.3**
	Supervised	74.5	67.8	3.1	**60.7**	61.7	82.1	44.9
	w/ Augmented Data	**87.3**	**86.7**	**0.8**	59	**67.3**	**92.5**	45.8
En–Pt	Vanilla	-	-	12.5	-	-	-	**44.6**
	Supervised	83.4	82.6	3.1	**60**	60.9	86.9	43.7
	w/ Augmented Data	**92.2**	**94.4**	**1.1**	59.5	**63.5**	**94.2**	44.1
En–Ru	Vanilla	-	-	5.3	-	-	-	25.6
	Supervised	70.6	54.5	2.4	**42**	39.5	83.7	**26.4**
	w/ Augmented Data	**80.7**	**77.2**	**1.5**	37.6	**39.8**	**91**	24.9
En–It	Vanilla	-	-	11.6	-	-	-	**27.9**
	w/ Augmented Data	93.7	89.4	**3.2**	53	50.9	94.6	27.6

Table 4: End-to-end MT model results. P and R denote precision and recall respectively.

LP	Direction	Model	P%	R%	F0.5
En–Es	M→F	GATE	**95**	40	0.75
		Ours	89.6	**69.2**	**0.85**
	F→M	GATE	**97**	50	0.82
		Ours	94.5	**73.7**	**0.89**
En–Fr	M→F	GATE	**91**	27	0.62
		Ours	89.3	**72.5**	**0.85**
	F→M	GATE	**97**	28	0.65
		Ours	96.1	**79.3**	**0.92**
En–It	M→F	GATE	**91**	32	0.66
		Ours	78.7	**58.8**	**0.74**
	F→M	GATE	**96**	47	0.79
		Ours	92	**75.1**	**0.88**

Table 5: Comparison of data augmentation pipeline using M2M against GATE on M → F and F → M re-writing. P and R denote precision and recall.

the setup and evaluation metrics used for this comparison can be found in Appendix I. We see significant improvements in recall at the cost of relatively small degradation in precision (except English-Italian). Our system is able to outperform GATE on their proposed F.5 metric on all 3 language pairs.

7.3 End-to-end MT model details

We train a vanilla multilingual MT model on all 6 language pairs using parallel corpora from Europarl, WikiMatrix, WikiTitles, Multi-UN (Chen and Eisele, 2012), NewsCommentary (Barrault et al., 2019) and Tilde MODEL (Rozis and Skadiņš, 2017). We refer to this as *vanilla bi-text*. We evaluate the models on gender-related metrics using our gender test set. The details of data pre-processing, training, and model architecture can be found in Appendix J.

A straightforward way to adapt this vanilla model to produce gender alternatives is to use domain adaptation methods towards the G-Trans dataset (which contains gender structures and alignments). To this end, we train another MT model with the *vanilla bi-text* plus the G-Trans dataset with a prefixed corpus tag `<gender>` using the loss and serialization described in section 4. Adding corpus tags when mixing corpora from different domains has proven to be quite effective (Kobus et al., 2017; Caswell et al., 2019; Costa-jussà et al., 2022). During inference, this tag is used to decode gender alternatives. We treat this model as the supervised baseline.

Finally, we train a third model, this time augmenting the entire *vanilla bi-text* with gender structures and alignments by passing it through our data augmentation pipeline (using M2M since running GPT at scale is cost-prohibitive).

To measure the impact of our approach on general domain translation performance, we evaluate the models on the FLoRes (Costa-jussà et al., 2022) test set. Since FLoRes references don't contain gender structures, we also remove gender structures from the outputs of our models (if any are present) while evaluating against FLoRes. We do so by choosing the gender form which is more probable according to the model: concretely, for every gender structure `BEG M MID F END`, we choose either `M` or `F` depending on which phrase has a higher average token log probability.

7.4 End-to-end MT model results

Table 4 summarizes the results of these models. The vanilla model cannot generate alternatives and shows a huge bias towards generating masculine forms (δ-BLEU ranging from 5.3 to 12.5 points). This bias is greatly reduced by the supervised baseline. The model trained on augmented data further reduces the bias and obtains the best performance in terms of alternative metrics, alignment accuracy, and δ-BLEU. This shows the effectiveness of the data augmentation pipeline. Augmented data also allows us to train a competitive system for English-Italian which lacks supervised data.

Results on general domain translation quality (Column FLoRes BLEU from Table 4) show that compared to the vanilla baseline, the model trained on augmented data suffers no degradation on English to German and Spanish and some degradations (-0.3 to -0.7 BLEU) on Engish to French, Portuguese, Russian and Italian.

8 Conclusion and Future Work

In this work, we study the task of generating entity-level alternatives when translating a sentence with gender ambiguities into a language with grammatical gender. We open source first train datasets, encouraging future research towards this task, and develop a data augmentation pipeline that leverages pre-trained MT models and LLMs to generate even larger train sets. Finally, we demonstrate that this data can be used effectively to train deployment-friendly MT models that generate alternatives without any additional inference cost or model components.

Our models and pipeline can enable new translation UIs that support fine-grained gender control and can also find applications in aiding human translators to automatically point out ambiguities and recommend alternative translations.

Future work includes exploring other genderless source languages apart from English (e.g., Chinese, Korean, and Japanese) and associated challenges, as well as extending the approach to non-binary and gender-neutral forms (Lardelli, 2023; Piergentili et al., 2023b; Savoldi et al., 2024).

Bias Statement

This work focuses on the bias a machine translation system can manifest by solely generating one translation from multiple valid ones that exist with respect to grammatical gender when translating from English to a more gendered language, e.g., French. Singling out one translation as such without offering users the ability to modify the output to match the grammatical gender the user intends for each entity causes two categories of harm: representation harm and quality-of-service harm (Madaio et al., 2020; Blodgett et al., 2020). It causes representational harm by reflecting the potential stereotypes that lead to the default translation (e.g., between occupations and gender) and quality-of-service harm by failing the users who need the output in the target language to be in a grammatical gender case other than what is generated by default. Our work advocates and proposes a solution for enabling users to choose from all equally correct translation alternatives.

Limitations

All mentions of "gender" in this work refer to the grammatical gender present in many languages of the world that are not genderless. Grammatical gender in linguistics is distinct from social gender: while grammatical gender is essentially a noun class system, the discussion surrounding social gender (male, female, nonbinary) encompasses a much more complex set of concepts, e.g., social constructs, norms, roles, and gender identities. Building effective solutions that facilitate inclusive conversations on these topics is not only an open problem in NLP, but many fields.

Moreover, the ambiguities in the linguistic grammatical gender are assumed to be, as in most of the gendered languages, binary: masculine and feminine. However, many languages have more grammatical genders (i.e., noun classes): e.g., Worrorra has masculine, feminine, terrestrial, celestial, and collective.

As such, our proposed resources, as presented so far, fall short of generating entity-level gender-neutral translations or disambiguation beyond the binary system of masculine/feminine. However, it's noteworthy that our pipeline, paired with suitable data resources, e.g., gender-neutral terms for lattice rescoring, forms a powerful instrument for addressing such more challenging settings.

Acknowledgements

We would like to thank Yi-Hsiu Liao, Hendra Setiawan, and Telmo Pessoa Pires for their contributions and discussions through different stages of the project, Matthias Sperber, António Luís Vilarinho dos Santos Lopes, and USC ISI's CUTELABNAME members for their constructive feedback on the paper drafts, and the whole Machine Translation team at Apple for their support for the project.

References

Bashar Alhafni, Nizar Habash, and Houda Bouamor. 2022. User-centric gender rewriting. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 618–631, Seattle, United States. Association for Computational Linguistics.

Loïc Barrault, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. 2019. Findings of the 2019 conference on machine translation (WMT19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 1–61, Florence, Italy. Association for Computational Linguistics.

Luisa Bentivogli, Beatrice Savoldi, Matteo Negri, Mattia A. Di Gangi, Roldano Cattoni, and Marco Turchi. 2020. Gender in danger? evaluating speech translation technology on the MuST-SHE corpus. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6923–6933, Online. Association for Computational Linguistics.

Su Lin Blodgett, Solon Barocas, Hal Daumé III, and Hanna Wallach. 2020. Language (technology) is power: A critical survey of "bias" in NLP. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5454–5476, Online. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Isaac Caswell, Ciprian Chelba, and David Grangier. 2019. Tagged back-translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 53–63, Florence, Italy. Association for Computational Linguistics.

Yu Chen and Andreas Eisele. 2012. MultiUN v2: UN documents with multilingual alignments. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2500–2504, Istanbul, Turkey. European Language Resources Association (ELRA).

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loïc Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No language left behind: Scaling human-centered machine translation. *CoRR*, abs/2207.04672.

Marta R. Costa-jussà, Carlos Escolano, Christine Basta, Javier Ferrando, Roser Batlle, and Ksenia Kharitonova. 2022. Interpreting gender bias in neural machine translation: Multilingual architecture matters. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(11):11855–11863.

Anna Currey, Maria Nadejde, Raghavendra Reddy Pappagari, Mia Mayer, Stanislas Lauly, Xing Niu, Benjamin Hsu, and Georgiana Dinu. 2022. MT-GenEval: A counterfactual and contextual dataset for evaluating gender accuracy in machine translation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4287–4299, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Edouard Grave, Michael Auli, and Armand Joulin. 2021. Beyond english-centric multilingual machine translation. *J. Mach. Learn. Res.*, 22(1).

Sarthak Garg, Stephan Peitz, Udhyakumar Nallasamy, and Matthias Paulik. 2019. Jointly learning to align and translate with transformer models. 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 4453–4462, Hong Kong, China. Association for Computational Linguistics.

Xinyang Geng and Hao Liu. 2023. Openllama: An open reproduction of llama.

Thamme Gowda, Zhao Zhang, Chris Mattmann, and Jonathan May. 2021. Many-to-English machine translation tools, data, and pretrained models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 306–316, Online. Association for Computational Linguistics.

Nizar Habash, Houda Bouamor, and Christine Chung. 2019. Automatic gender identification and reinflection in Arabic. In *Proceedings of the First Workshop on Gender Bias in Natural Language Processing*, pages 155–165, Florence, Italy. Association for Computational Linguistics.

Melvin Johnson. 2020. A scalable approach to reducing gender bias in google translate. `https://blog.research.google/2020/04/a-scalable-approach-to-reducing-gender.html`. Accessed: 2024-01-28.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Catherine Kobus, Josep Crego, and Jean Senellart. 2017. Domain control for neural machine translation. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 372–378, Varna, Bulgaria. INCOMA Ltd.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of Machine Translation Summit X: Papers*, pages 79–86, Phuket, Thailand.

James Kuczmarski and Melvin Johnson. 2018. Gender-aware natural language translation.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia. Association for Computational Linguistics.

Manuel Lardelli. 2023. Gender-fair translation: a case study beyond the binary. *Perspectives*, 0(0):1–17.

Michael A. Madaio, Luke Stark, Jennifer Wortman Vaughan, and Hanna Wallach. 2020. Co-designing checklists to understand organizational challenges and opportunities around fairness in ai. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, page 1–14, New York, NY, USA. Association for Computing Machinery.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. 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. Association for Computational Linguistics.

Andrea Piergentili, Dennis Fucci, Beatrice Savoldi, Luisa Bentivogli, and Matteo Negri. 2023a. Gender neutralization for an inclusive machine translation: from theoretical foundations to open challenges. In *Proceedings of the First Workshop on Gender-Inclusive Translation Technologies*, pages 71–83, Tampere, Finland. European Association for Machine Translation.

Andrea Piergentili, Beatrice Savoldi, Dennis Fucci, Matteo Negri, and Luisa Bentivogli. 2023b. Hi guys or hi folks? benchmarking gender-neutral machine translation with the GeNTE corpus. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14124–14140, Singapore. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Spencer Rarrick, Ranjita Naik, Varun Mathur, Sundar Poudel, and Vishal Chowdhary. 2023. Gate: A challenge set for gender-ambiguous translation examples. In *Proceedings of the 2023 AAAI/ACM Conference on AI, Ethics, and Society*, AIES '23, page 845–854, New York, NY, USA. Association for Computing Machinery.

Adithya Renduchintala, Denise Diaz, Kenneth Heafield, Xian Li, and Mona Diab. 2021. Gender bias amplification during speed-quality optimization in neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 99–109, Online. Association for Computational Linguistics.

Roberts Rozis and Raivis Skadiņš. 2017. Tilde MODEL - multilingual open data for EU languages. In *Proceedings of the 21st Nordic Conference on Computational Linguistics*, pages 263–265, Gothenburg, Sweden. Association for Computational Linguistics.

Eduardo Sánchez, Pierre Andrews, Pontus Stenetorp, Mikel Artetxe, and Marta R. Costa-jussà. 2023. Gender-specific machine translation with large language models. *CoRR*, abs/2309.03175.

Danielle Saunders and Bill Byrne. 2020. 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. Association for Computational Linguistics.

Danielle Saunders, Rosie Sallis, and Bill Byrne. 2022. First the worst: Finding better gender translations during beam search. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3814–3823, Dublin, Ireland. Association for Computational Linguistics.

Beatrice Savoldi, Andrea Piergentili, Dennis Fucci, Matteo Negri, and Luisa Bentivogli. 2024. A prompt response to the demand for automatic gender-neutral translation. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 256–267, St. Julian's, Malta. Association for Computational Linguistics.

Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. 2021. WikiMatrix: Mining 135M parallel sentences in 1620 language pairs from Wikipedia. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1351–1361, Online. Association for Computational Linguistics.

Artūrs Stafanovičs, Toms Bergmanis, and Mārcis Pinnis. 2020. Mitigating gender bias in machine translation with target gender annotations. In *Proceedings of the Fifth Conference on Machine Translation*, pages 629–638, Online. Association for Computational Linguistics.

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

Tony Sun, Andrew Gaut, Shirlyn Tang, Yuxin Huang, Mai ElSherief, Jieyu Zhao, Diba Mirza, Elizabeth Belding, Kai-Wei Chang, and William Yang Wang. 2019. Mitigating gender bias in natural language processing: Literature review. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1630–1640, Florence, Italy. Association for Computational Linguistics.

Tony Sun, Kellie Webster, Apurva Shah, William Yang Wang, and Melvin Johnson. 2021. They, them, theirs: Rewriting with gender-neutral english. *CoRR*, abs/2102.06788.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey. European Language Resources Association (ELRA).

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971.

Eva Vanmassenhove, Chris Emmery, and Dimitar Shterionov. 2021. NeuTral Rewriter: A rule-based and neural approach to automatic rewriting into gender neutral alternatives. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8940–8948, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Tatu Ylonen. 2022. Wiktextract: Wiktionary as machine-readable structured data. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1317–1325, Marseille, France. European Language Resources Association.

A Dataset details

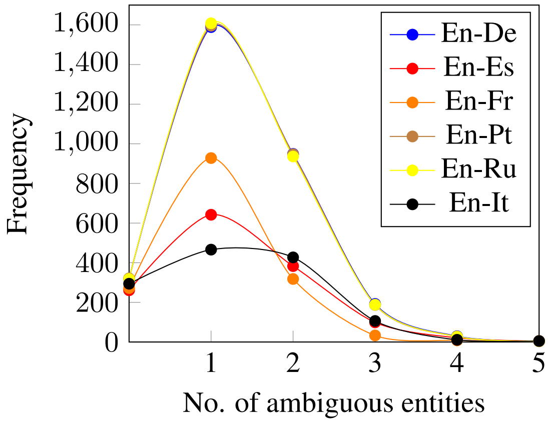

Figure 2: Number of examples v.s. number of ambiguous entities in the test set.

Detailed train data statistics are listed in Table 6. Detailed test set statistics are shown in Table 7 and Figure 2.

We had to get the annotations in GATE and MT-GenEval reviewed and post-edited from human annotators because their annotation guidelines differ from ours in the following respects:

- GATE defines a gender-ambiguous entity as an entity whose gender cannot be inferred from the grammatical sentence context *and* whose gender can influence changes in the translation. This second requirement makes this definition of *ambiguous entity* dependent on the target language/translation. E.g., in “I am going to the market”, despite the gender of *I* being ambiguous, it would not be marked as an ambiguous entity for English-Spanish, since the Spanish translation does not change based on the gender of *I*. The same entity would be marked as an ambiguous entity in case of English-Hindi where the translation changes based on the gender of *I*.

  In our definition of an ambiguous entity, we drop the second requirement, making it independent of the translation and the target language. This enables us to train an ambiguity tagger solely on the Engish source sentences which can be used for any English-X language pair. This, however, forces us to re-annotate the GATE corpus.

- MT-GenEval corpus contains source sentences with annotated entities whose gender *can* be inferred as masculine/feminine from the sentence context. This provides a valuable test-bed for catching false positive gender alternatives. However, we found that $\sim 50\%$ of source sentences also contain one or more ambiguous entities which have not been annotated. Therefore we re-annotate the MT-GenEval corpus as well to mark such entities.

Upon the deanonymized publication of this work, we plan to release the datasets under CC BY-SA license.

B Problem of masculine generics during gender ambiguity annotation

It is fairly common to use masculine gendered words to refer to ambiguous entities. In administrative and legal text, masculine gendered words have been used to refer to collection of people (Piergentili et al., 2023a) for e.g. “A judge must certify that **he** has familiarized **himself** with...”. It is a complex problem to ascertain whether *he* refers to a masculine individual or a group of (ambiguous gendered) people at large.

In our annotation guidelines we informed the annotators that entities shouldn’t be marked as masculine solely because of masculine generic nouns like *actor*, *sportsmen*. However no special guidelines were provided around the trickier case of masculine generic pronouns (*he*, *himself* as shown in the example above)

C Synthetically generated train data

We used human annotation to collect primary versions of G-Trans and G-Tag datasets (gender train sets) using the annotation process described in subsection 3.1. However, we are unable to release these “human-annotated” sets publicly due to legal and proprietary data restrictions. To make our approach and results reproducible to the community, we instead plan to release "synthetically generated sets" generated as follows: we trained our data augmentation pipeline (described in section 5) on the “human-annotated” training sets and then ran the data augmentation pipeline on corpora mentioned subsection 3.2. We then sampled the G-Trans and G-Tag datasets from the pipeline results and use them throughout our work.

D Gender-ambiguous entity detector

The gender-ambiguous entity detector is fine-tuned using the following hyper-parameters:

Dataset	Statistic	En-De	En-Es	En-Fr	En-Pt	En-Ru
G-Tag	Sentences	11.7	13.5	13.3	13.3	10.3
	Ambiguous entities	13.8	14	13.2	14.6	11.3
	Masculine entities	7.4	7.8	7.6	7.9	6.6
	Feminine entities	6.1	7	6.7	7	5.6
G-Trans	Sentences	49.4	49.6	49.7	49.6	48.8
	Ambiguous entities	69.3	74.7	69.1	73.9	64.1
	Gender structures	77.5	81.7	77.6	83.1	72.5

Table 6: Train set statistics: All numbers are *in thousands*. We sample about 12k sentences for the G-Tag dataset, roughly containing $2:1:1$ ratio of ambiguous, masculine and feminine entities. About 50k sentence pairs with ambiguous entities and gender structures are sampled for the G-Trans dataset.

Language Pair	No. of sentences with		
	Total	1+ Ambiguous entities	1+ gender structures
En-De	3038	2765	2118
En-Es	1407	1147	972
En-Fr	1564	1292	1006
En-Pt	3083	2764	2435
En-Ru	3083	2765	1847
En-It	1312	1018	858

Table 7: Test set statistics: About $80-90\%$ sentences contain at least one gender-ambiguous entity, out of which about $60-80\%$ contain gender structures in the reference.

- `batch size: 64`
- `epochs: 2`
- `learning rate: 2e-5`
- `tokenizer: intl from sacrebleu library`
- `subword model: default xlm-roberta-large tokenizer`
- `output labels:` `<A>` (ambiguous), `<M>` (masculine), `<F>` (feminine), `<N>` (not an entity)
- `linear tagging layer:` 1024×4
- Architecture hyper-parameters can be found by loading `xlm-roberta-large` using `AutoModelForTokenClassification` in `transformers`.
- The tagging loss is applied only on the first sub-word of each token. The prediction for each token is computed based on the label output for the first sub-word.
- We fine-tune all the parameters of the pre-trained model along with the added linear layer.
- All reported results are gathered from a single run.

Table 8 summarizes the results of the detector on tagging entities of different genders.

E Generating all-masculine/feminine translations by finetuned-M2M model

We fine-tuned a pre-trained `M2M-1.2B` model with the following hyper-parameters:

- `batch size: 8192`
- `learning rate: 3e-5`
- `encoder layerdrop: disabled`
- `decoder layerdrop: disabled`
- Rest of the hyper-parameters are the same as the pre-trained model.
- We fine-tune for a total of `40000` steps and select the best checkpoint based on loss on a held out validation set.
- We use the sub-word model and dictionaries of the pre-trained M2M model. However, we add gender assignment tags (`<M>` and `<F>`) as new entries in the dictionary and train their embeddings from scratch.
- We use a beam size of `5` while decoding all-masculine/feminine translations using lattice-rescoring.
- All reported results are gathered from a single run.

F Ablation studies on generating using LLMs

We study the effect of three factors on the effectiveness of LLMs for generating all-masculine/feminine translations as part of our data augmentation process: number of in-context examples, prompt design, and choice of LLM.

F.1 Number of in-context examples

In our preliminary experiments, we found using at least four in-context examples to be necessary for our task, with performance starting to plateau thereafter (see the chart below in Figure 3). We use six in-context examples in the rest of the experiments.

Language Pair	Ambiguous Entities Precision%	Ambiguous Entities Recall%	Masculine Entities Precision%	Masculine Entities Recall%	Feminine Entities Precision%	Feminine Entities Recall%
En-De	93.1	91.4	72.5	83.0	74.7	84.2
En-Es	89.8	86.6	70.3	82.3	74.8	83.6
En-Fr	90.3	88.1	69.0	80.0	70.0	80.7
En-Pt	93.1	91.4	70.6	84.4	73.2	87.8
En-Ru	93.2	91.3	71.7	83.9	73.6	84.0
En-It	92.1	89.2	72.3	84.4	72.0	85.7

Table 8: Results of tagging different gendered entities by the `XLM` based tagger.

Language Pair	LLM	Prompting View	Alternatives Metrics ↑ Precision%	Alternatives Metrics ↑ Recall%	Structure Metrics↑ Precision%	Structure Metrics↑ Recall%
En–De	GPT	Generator	**91.5**	81.8	73.2	74.8
		Editor	89.4	**86.1**	**73.9**	**76**
	OpenLLaMA	Generator	91.5	26.6	**48.2**	**41.4**
		Editor	**92.5**	**47.8**	43.4	37.6
En–Es	GPT	Generator	90.3	87.9	60.4	66.4
		Editor	**91.6**	**92.4**	**63.5**	**69.5**
	OpenLLaMA	Generator	67.5	7.9	31.1	26.9
		Editor	**91.4**	**34**	**52.9**	**40.7**
En–Fr	GPT	Generator	87.4	82.3	**69.4**	**77**
		Editor	**88.1**	**86.8**	63.8	75.7
	OpenLLaMA	Generator	54.6	5.3	24.7	28
		Editor	**85.9**	**32.8**	**58.4**	**52.3**
En–Pt	GPT	Generator	**94**	78.1	**66**	**66.8**
		Editor	92.8	**79.8**	63.3	66.6
	OpenLLaMA	Generator	89.7	11.8	46.6	32
		Editor	**93.7**	**44.8**	**54**	**43.6**
En–Ru	GPT	Generator	**83.9**	**61.8**	45.9	45.1
		Editor	80.1	55.3	**48.8**	**49.4**
	OpenLLaMA	Generator	67.6	6.4	8.9	8.4
		Editor	**79.1**	**12.1**	**27.4**	**21.4**

Table 9: LLM Ablation Results.

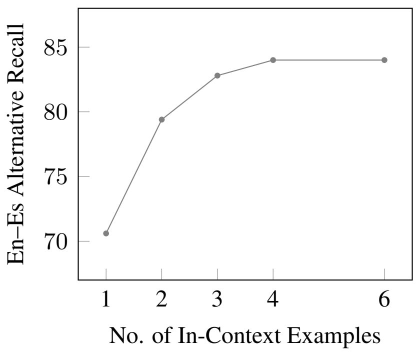

Figure 3: Ablation on the number of in-context examples. We use the GPT's alternative recall on English–Spanish as an exemplar. Per this results, we use six in-context examples for prompting.

F.2 Choice of LLM and prompt design

In addition to GPT (`gpt-3.5-turbo`), we also experiment with OpenLLaMA (`OpenLlama-v2-7B`) (Geng and Liu, 2023), an open reproduction of LLaMA (Touvron et al., 2023). We find these two to vary in overall performance and robustness to different kinds of prompts.

Specifically, besides the prompt design discussed in the main text, which has the LLM *edit* an existing translation to satisfy the provided grammatical gender requirements, we also experiment with an additional design: given the input and the grammatical gender requirements, we have the LLM generate the translation from scratch (Figure 4). We call the former the editor-view prompting, and the latter the generator-view prompting.

In editor-view prompting, the base translation can be sourced in any number of ways, including using the reference translation, as we did in subsection 7.2. However, to make the study between editor-view and generator-view fair and make sure reference translations do not give any advantage to the editor-view, we first prompt the LLM for base translations (first call) and then have it edit those (second call). This effectively breaks the task of generating gender alternatives down to two separate tasks for LLMs: translation, and then editing.

In-Context Example(s)

English input: That is not what the taxpayers and consumers of Europe want.
Spanish translation (feminine taxpayers, feminine consumers): No es eso lo que quieren las contribuyentes y las consumidoras de Europa.
Spanish translation (masculine taxpayers, masculine consumers): No es eso lo que quieren los contribuyentes y los consumidores de Europa.
...

Test Input

English input: $\{x\}$
Spanish translation ($\{$feminine i $\forall i \in G_a\}$):

LLM Output

$\{y_F\}$
Spanish translation ($\{$masculine i $\forall i \in G_a\}$): $\{y_M\}$

Figure 4: Prompting LLMs using in-context examples to generate translations with all-masculine and all-feminine gender assignments from scratch.

Table 9 reports and compares the results of prompting each of the two LLMs we experiment with, using each of the two prompt designs we use. All reported results are gathered from a single run. GPT, expectedly, outperforms OpenLLaMA. And while both generally benefit from breaking down the task under the editor-view (and perform better under editor-view than under generator-view), OpenLLaMA conspicuously profits more. Specifically, OpenLLaMA's alternative recall under the generator-view suggests that it fails to generate alternatives following the in-context examples. However, under the editor-view, it is able to follow the in-context examples more. The wider gap between the performance of OpenLLaMA under the two prompting approaches compared to that of GPT, shows that for our task, it's far less robust to different prompt designs.

G Aligning gender-ambiguous entities

We fine-tune an `xlm-roberta-large` model for aligning gender structures to their corresponding ambiguous entities using the following hyper-parameters:

- `epochs: 1`
- `output labels: 1`(aligned), `2` (not-aligned)
- `linear tagging layer:` 1024×2
- Rest of the hyper-parameters are same as the gender-ambiguous entity detector (Appendix D).
- All reported results are gathered from a single run.

Figure 5 shows an example of input and output when aligning a gender structure.

H Running data augmentation pipeline on outputs of M2M and GPT

In this work we focus on running the data augmentation pipeline over parallel corpora to enrich them with gender structures and gender alignments. However, the pipeline can also be run over *any translation* system to generate entity-level gender alternatives. Table 10 shows the results when the data augmentation pipeline is run over translations from the pre-trained M2M and GPT models.

The pipeline uses fine-tuned M2M when run over translations from the M2M model and the editor-view prompting using GPT when run over translations from GPT. We can see that both M2M and GPT have large bias towards producing masculine translations (δ-BLEU values ranging from 6.5 to 12.7 points). The data augmentation pipeline has multiple components and much higher inference cost than the end-end student model, but can produce higher quality gender alternatives when compared to the end-end model (Table 4 vs. Table 10).

I Comparison against GATE

For the comparison against GATE in Table 5, we use exactly the same setup and metrics (Precision/Recall/F0.5) from Rarrick et al. (2023). We evaluate our data augmentation pipeline on the gender re-writing task. Let's consider the `M` $\rightarrow$ `F` re-writing case: Given a source sentence with ambiguous entities, the task is to re-write an all-masculine reference translation into an all-feminine reference translation. A system might not output a re-write (in case it fails to detect any ambiguous entities or if the re-written output is the same as the input) or it might actually do a re-write. If the system performs a re-write, it's classified as *correct* if the re-write matches the all-feminine reference translation exactly. If there is any difference between the two, then the re-write is classified as *incorrect*. Given these definitions, the Precision and Recall is defined as:

$$\text{Precision} = \frac{\text{number of correct re-writes}}{\text{number of attempted re-writes}}$$

$$\text{Recall} = \frac{\text{number of correct re-writes}}{\text{total number of examples}}$$

J End-to-end MT model to generate alternatives

We extract the bi-text used for training end-to-end models using `mtdata` (Gowda et al., 2021). We

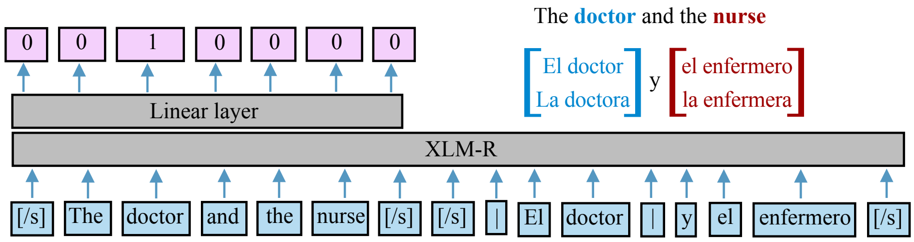

Figure 5: This figure shows an example of aligning the gender structure (El doctor / La doctora). The model is fine-tuned to classify the source tokens as being aligned (1) or not-aligned (0) to this gender structure.

Language Pair	Model	Alternatives Metrics↑ Precision%	Alternatives Metrics↑ Recall%	BLEU Masc.↑	BLEU Fem.↑	BLEU $\delta\downarrow$	Structure Metrics↑ Precision%	Structure Metrics↑ Recall%
En–De	M2M	-	-	46.8	36.6	10.2	-	-
	+ Data Augmentation	**92.5**	82.8	46.9	45.7	1.2	64.7	64.2
	GPT	-	-	**53.8**	41.4	12.4	-	-
	+ Data Augmentation	89.4	**86.1**	**53.8**	**52.7**	**1.1**	**73.9**	**76**
En–Es	M2M	-	-	47.3	37	10.3	-	-
	+ Data Augmentation	**95.8**	91.3	47.5	46.5	**1**	63.2	64
	GPT	-	-	**51.8**	40.4	11.4	-	-
	+ Data Augmentation	91.6	**92.4**	51.5	**50.4**	1.1	**63.5**	**69.5**
En–Fr	M2M	-	-	50	41.5	8.5	-	-
	+ Data Augmentation	**90.7**	84	52.4	48.8	3.6	54.5	67.6
	GPT	-	-	**58.5**	48.4	10.1	-	-
	+ Data Augmentation	88.1	**86.8**	58.3	**57**	**1.3**	**63.8**	**75.7**
En–Pt	M2M	-	-	49.2	36.9	12.3	-	-
	+ Data Augmentation	**94.1**	**94.2**	49.2	48.3	**0.9**	59.1	60.1
	GPT	-	-	54.1	40.6	13.5	-	-
	+ Data Augmentation	92.8	79.8	**54.2**	**51.5**	2.7	**63.3**	**66.6**
En–Ru	M2M	-	-	29.2	22.7	6.5	-	-
	+ Data Augmentation	**86.9**	**81.1**	29.3	27.9	**1.4**	44.6	42.3
	GPT	-	-	**31.8**	24.1	7.7	-	-
	+ Data Augmentation	80.1	55.3	31.3	**28.2**	3.1	**48.8**	**49.4**
En–It	M2M	-	-	46.8	34.1	12.7	-	-
	+ Data Augmentation	95.9	84.6	**47**	**43.3**	**3.7**	54.7	50.9

Table 10: Results of the data augmentation pipeline applied to vanilla translations produced by pre-trained M2M and GPT models.

use `sentencepiece` (Kudo, 2018) to learn a vocabulary of size 36000 tokens. We remove sentence pairs with lengths ≥ 400 sentencepiece tokens or exceeding a token ratio of $1: 3$. We train all end-to-end models using the following hyper-parameters:

- `batch size: 458752`
- `decoder layers: 20`
- `decoder layers: 3`
- `lr: 7e-4`
- We supervise an attention head in second from the bottom decoder layer. The scaling factor λ for the alignment loss is set to `0.05`.
- `embedding dim: 512`
- shared encoder-decoder and input-output embeddings
- `learning rate: 3e-5`
- All reported results are gathered from a single run.

The end-end models produce gender structures without any constraints. This can result in gender structures containing phrases that differ in more than just gender inflections. To avoid this, we explicitly check the gender structures against our collected list of gender inflections and retain only those structures which pass the check.

K Evaluation Metrics

The alternatives metrics compute the sentence level precision and recall of generating alternatives. Let $\mathbf{I}(b)$ denote an indicator function:

$$\mathbf{I}(b) = \begin{cases} 1 & b = \text{True} \\ 0 & b = \text{False} \end{cases}$$

and given a sentence x, let $\phi(x)$ check whether x contains gender structures:

$$\phi(x) = \begin{cases} \text{True} & x \text{ contains gender structures} \\ \text{False} & \text{otherwise} \end{cases}$$

Let y and $\hat{y}$ denote the reference from the test set and the system hypothesis respectively, then alternatives precision and recall can be defined as follows:

$$\text{Precision} = \frac{\sum_{y,\hat{y}} \mathbf{I}(\phi(y) \wedge \phi(\hat{y}))}{\sum_{\hat{y}} \mathbf{I}(\phi(\hat{y}))}$$

$$\text{Recall} = \frac{\sum_{y,\hat{y}} \mathbf{I}(\phi(y) \wedge \phi(\hat{y}))}{\sum_{y} \mathbf{I}(\phi(y))}$$

We compute structure metrics over the subset S where both references and system outputs contain gender structures, i.e. $S = \{(y, \hat{y}) \mid \phi(y) \wedge \phi(\hat{y}) = \text{True}\}$. Over S, we compute the following statistics:

- Total structures: total number of gender structures present in y for $(y, \hat{y}) \in S$.
- Predicted structures: total number of gender structures present in $\hat{y}$ for $(y, \hat{y}) \in S$
- Correct structures: total number of gender structures which are present in both y and $\hat{y}$ for $(y, \hat{y}) \in S$

We can then compute structure precision and recall as follows:

$$\text{Precision} = \frac{\text{Correct structures}}{\text{Predicted structures}}$$

$$\text{Recall} = \frac{\text{Correct structures}}{\text{Total structures}}$$