Fairness for Text Classification Tasks with Identity Information Data Augmentation Methods

Mohit Wadhwa, Mohan Bhambhani, Ashvini Jindal, Uma Sawant, Ramanujam Madhavan
LinkedIn Corporation
(mwadhwa,mbhambhani,ajindal,usawant,rmadhavan)@linkedin.com

ABSTRACT
Counterfactual fairness methods address the question: How would the prediction change if the sensitive identity attributes referenced in the text instance were different? These methods are entirely based on generating counterfactuals for the given training and test set instances. Counterfactual instances are commonly prepared by replacing sensitive identity terms, i.e., the identity terms present in the instance are replaced with other identity terms that fall under the same sensitive category. Therefore, the efficacy of these methods depends heavily on the quality and comprehensiveness of identity pairs. In this paper, we offer a two-step data augmentation process where (1) the former stage consists of a novel method for preparing a comprehensive list of identity pairs with word embeddings, and (2) the latter consists of leveraging prepared identity pairs list to enhance the training instances by applying three simple operations (namely identity pair replacement, identity term blindness, and identity pair swap). We empirically show that the two-stage augmentation process leads to diverse identity pairs and an enhanced training set, with an improved counterfactual token-based fairness metric score on two well-known text classification tasks.

KEYWORDS
fairness, counterfactuals, word embeddings

1 INTRODUCTION
Machine learning models learn the statistical patterns from historical instances to predict labels for future instances. These models are intended to learn bias; for example, a toxic text classifier is expected to understand the presence of toxic terms and assign a higher score to those terms compared to other non-toxic terms. However, models are not intended to discriminate between two instances based on sensitive identity terms, such as race or gender indication terms [1, 3, 14]. Dixon et al. [6] showed that due to the disproportionate distribution of terms in the training data, text classification models could unintentionally learn bias (defined as unintended bias) specific to the identity terms.

At the instance level, a related bias issue arises when a classifier assigns different scores to two nearly identical instances with different identity information. Garg et al. [7] study counterfactual fairness by considering the question: How would the prediction change if the sensitive identity attributes referenced in the text instance were different? To address counterfactual fairness, it compares three approaches to address instance-level bias and introduces a counterfactual token fairness metric. Both measurement and mitigation methods discussed depend heavily on the counterfactual instances generation process. Counterfactual instances are generated by replacing identity terms present in the instance with other identity terms that fall in the same category. For example, replacing the term women in the sentence “women should rule the world” with the term men to create a counterfactual sentence “men should rule the world”. Counterfactual fairness methods, thus, demand information about identity pairs (such as women-men, gay-straight), and could extensively benefit from diverse identity pairs information and counterfactual generation mechanisms. However, Garg et al. [7] limits the process to a manually prepared list of 50 identity pairs.

In this paper, we present Identity Information Data Augmentation methods, IIDA for brevity, to bolster the text classification models against unintended bias by automatically generating diverse counterfactual instances. Given a subset of known identity pairs [4, 7], we first apply a simple but powerful method to augment the identity pairs with additional multifarious identity pairs that occur in the same context. We then use the expanded comprehensive identity pairs list to measure the counterfactual token fairness metric. Prepared identity pairs along with a set of instance perturbations (such as blindness, replacement, and swap) are used to produce counterfactual instances that are leveraged for training the model either in the form of augmentation or as a logit-pairing training scheme [7, 8]. In summary, the main contributions of our paper are:

1) Identity Pair Augmentation (Section 3.1): We propose a novel word-embeddings based method to generate diverse identity pairs, as shown in Figure 1a. Diverse identity pairs are leveraged for preparing counterfactual training instances and for measuring counterfactual token fairness [7].

2) Identity Information Perturbations (Section 3.2): To enhance the training set with counterfactuals information, we leverage augmented identity pairs to perform three types of
We present the identity pair augmentation procedure in Algorithm 1. We propose a two-step data augmentation process where we first extract top \( k \) most similar words for identity terms \( i_1 \) and \( i_2 \), and form a cartesian product (line 3) of the two extracted words set denoted by \( J \). \( J \) captures the set of word pairs that possibly occur in the same context with respect to the input pair, and to filter out the pairs that don’t occur in the same context (i.e., to remove noisy pairs), we perform the steps presented in lines 5-9. For each pair \( (j_1, j_2) \), we extract top \( k_2 \) most similar words for identity terms \( j_1 \) and \( j_2 \) and compute a similarity score (with sets similarity methods like Jaccard-index) between the two sets. We consider \( (j_1, j_2) \) a valid identity pair only if the similarity score computed before is above a selected threshold value \( \tau \). Augmented identity terms are simply a set of unique terms that are part of augmented identity pairs. We present the parameter values in the experiment section below and show that the proposed procedure works well in practice.

We also investigated the word embedding model with analogy operation where given input term1 and term2, we iterate over \( top_k \) most similar terms/words for term1, and we represent each similar term by term1'. We then find analogy term2' with the following setting:

\[
w(\text{term2}') = w(\text{term2}) + w(\text{term1}') - w(\text{term1})
\]

We considered \( (\text{term1}', \text{term2}') \) as an augmented identity pair. We found that even a single generic term in \( top_k \) terms could lead to invalid out-of-context identity pairs. For example - say we consider \( \text{man, woman} \) as a known identity pair, then if term ‘person’ occurs in \( top_{10} \) similar words to term ‘man’, finding analogy would lead to erroneous terms like ‘individual’,’each’, ‘whether’, ‘every’.

Algorithm 1 we propose avoids this by doing a similarity check over \( top_{k2} \) similar terms and using a similarity threshold to avoid invalid pairs. We observed that in practice the identity pairs generated by Algorithm 1 are diverse and valid in most cases. We share a sampled list of the identity pairs generated by Algorithm 1 in Appendix B section.

### 3.2 Identity Information Perturbations

We here present the details to leverage augmented identity terms and pairs for generating counterfactual instances. We additionally perform two other operations, blindness, and swap, for instance perturbation. On each instance in the training set, we perform the following operations:

#### Algorithm 1 Identity Pair Augmentation

**Notations:**
- \( \zeta \) - function to get top-k most similar words
- \( \psi \) - function to get similarity score

**Input:**
- \( I \) - known identity pairs
- \( M \) - word embedding model
- \( \tau \) - similarity score threshold
- \( top_{k1} \) - top-k value for augmentation
- \( top_{k2} \) - top-k value for similarity check

**Output:** \( O \) - augmented identity pairs

**Process:**

1. \( O := \text{set}(I) \)
2. \( \forall (i_1, i_2) \in I \)
3. \( J := \zeta(M, i_1, top_{k1}) \times \zeta(M, i_2, top_{k1}) \)
4. \( \forall (j_1, j_2) \in J \)
5. \( s_1 := \zeta(M, j_1, top_{k2}) \)
6. \( s_2 := \zeta(M, j_2, top_{k2}) \)
7. if \( \psi(s_1, s_2) \geq \tau \) do
8. \( O := add((j_1, j_2)) \)
9. end
10. return \( O \)

Table 1: Identity Pairs

| Known Pair       | Other Possible Pairs                                      |
|------------------|----------------------------------------------------------|
| man-woman        | he-she, boy-girl, guy-wife, boy-person, him-girl, father-mother |
| blacks-whites    | black-american, latino-mexican, hispanic-american, mexican-american, hispanic-american |
| trump-biden      | trump-obama, palin-mitt, barack-clinton, palin-hillary, trump-hillary |

(3) **Effectiveness (Section 4):** We methodically evaluate IIDA on two well-recognized unintended bias inducing text classification tasks, demonstrating that IIDA methods help mitigate the unintended bias while sustaining the model performance on both the tasks.

2 RELATED WORK

Garg et al. [7] propose a fairness metric, Counterfactual Token fairness (CTF gap), for measuring model’s bias where bias is computed by taking the absolute difference between model’s prediction score on instance and its counterfactual, averaged over all records. Blindness, Counterfactual Augmentation, and Counterfactual Logit Pairing methods are also discussed by [7] for optimizing counterfactual fairness while training text classification models. Qian et al. [13] edit the loss function to equalize the probabilities of male and female words to alleviate bias in word-level language models. Wei and Zou [15] presents data perturbation techniques for boosting performance of text classification models. Synonym replacement, random insertion/swap/deletion, perturbation operations are performed by Wei and Zou [15]. Our work leverages the perturbation operations thought to bolster fairness metric.

3 IIDA

We propose a two-step data augmentation process where we first augment known identity pairs with diverse identity pairs to make it comprehensive. We then apply perturbation operations to enhance the training set and for the counterfactual token fairness metric. Table 1 presents a sampled set of known identity pairs and other possible diverse pairs.

### 3.1 Identity Pair Augmentation

We present the identity pair augmentation procedure in Algorithm 1. Given a set of known identity pairs \( I \), we leverage the word embedding model [2, 10, 11], to augment identity pairs. For each known identity pair \( (i_1, i_2) \), we use embedding model \( M \) to extract top \( top_{k1} \) most similar words for identity terms \( i_1 \) and \( i_2 \), and form a cartesian product (line 3) of the two extracted words set denoted by \( J \). \( J \) captures the set of word pairs that possibly occur in the same context with respect to the input pair, and to filter out the pairs that don’t occur in the same context (i.e., to remove noisy pairs), we perform the steps presented in lines 5-9. For each pair \( (j_1, j_2) \), we extract top \( top_{k2} \) most similar words for identity terms \( j_1 \) and \( j_2 \), and compute a similarity score (with sets similarity methods like Jaccard-index) between the two sets. We consider \( (j_1, j_2) \) a valid identity pair only if the similarity score computed before is above a selected threshold value \( \tau \). Augmented identity terms are simply a set of unique terms that are part of augmented identity pairs. We present the parameter values in the experiment section below and show that the proposed procedure works well in practice.

We also investigated the word embedding model with analogy operation where given input term1 and term2, we iterate over \( top_k \) most similar terms/words for term1, and we represent each similar term by term1'. We then find analogy term2' with the following setting:

\[
w(\text{term2}') = w(\text{term2}) + w(\text{term1}') - w(\text{term1})
\]

We considered \( (\text{term1}', \text{term2}') \) as an augmented identity pair. We found that even a single generic term in \( top_k \) terms could lead to invalid out-of-context identity pairs. For example - say we consider \( \text{man, woman} \) as a known identity pair, then if term ‘person’ occurs in \( top_{10} \) similar words to term ‘man’, finding analogy would lead to erroneous terms like ‘individual’, ‘each’, ‘whether’, ‘every’.

Algorithm 1 we propose avoids this by doing a similarity check over \( top_{k2} \) similar terms and using a similarity threshold to avoid invalid pairs. We observed that in practice the identity pairs generated by Algorithm 1 are diverse and valid in most cases. We share a sampled list of the identity pairs generated by Algorithm 1 in Appendix B section.

### 3.2 Identity Information Perturbations

We here present the details to leverage augmented identity terms and pairs for generating counterfactual instances. We additionally perform two other operations, blindness, and swap, for instance perturbation. On each instance in the training set, we perform the following operations:
A **Identity Pair Replacement (IPR):** Randomly sample $n_r$ identity pairs from the augmented identity pairs and replace all the first term occurrences with the second term of the pair.

B **Identity Term Blindness (ITB):** Randomly sample $n_s$ identity terms from the augmented identity terms and replace all the occurrences of the sampled terms with the blindness placeholder. Each term could have a dedicated placeholder; for example, a gender representing term could have a placeholder ‘GENDER_TOKEN’, and similarly, ‘NAME_TOKEN’ for a public figure name.

C **Identity Pair Swap (IPS):** Randomly sample $n_s$ identity pairs from the augmented identity pairs and swap all the sampled pairs’ occurrences.

We selected these operations based on the research by Wei and Zou [15], and heuristics that counterfactual and swap instances help classifier learn and optimize diverse data while blindness operations help hide sensitive information specific to categories with several sub-categories (e.g., public figure names). IPR operation on an input instance $x$, leads to the generation of diverse counterfactual instances denoted by $\phi_r(x)$. Similarly, ITB operation on $x$ leads to the generation of blindness instances represented by $\phi_b(x)$, and the collection of instances generated by IPS operation are characterized by $\phi_s(x)$. IPR and ITB operations are also considered by [7] but with diverse identity pairs it helps to mitigate bias, as we demonstrate in Section 4.

We further leverage $\phi_s$ and $\phi_b$ to update the logit pairing training scheme proposed by [7] to the following objective:

$$
\sum_{x \in X} C(f(x), y) + \lambda_1 \sum_{x \in X} E_{x_r \sim U[nf]} | g(x) - g(x_r) | + \lambda_2 \sum_{x \in X} E_{x_b \sim U[nf]} | g(x) - g(x_b) | 
$$

where $X, C, f, y$ denote the training set, cost function, classifier, and true prediction value respectively, $g$ produces a logit, and $\lambda_1, \lambda_2$ are penalty hyperparameters. Counterfactual and blinded instances individually have shown to mitigate bias [7], and we experiment the logit pairing training scheme with a combination of both settings. Both settings act complementary, i.e., sensitive categories with limited list of identity pairs (for example, race sensitive category) could be handled by counterfactual instances, and categories with exhaustive list of identity pairs (for example, public figure names category) could be handled by blinded instances. Swap instances $\phi_s$ could also be considered with a logit pairing scheme, but we keep it out of paper scope.

## 4 EXPERIMENTS

### 4.1 Benchmark Datasets

We perform experiments on two public text classification benchmark datasets: (1) Toxic Comment Classification Dataset ¹, (2) Hate Speech Dataset ² [5]. We combine six classes (toxic, severe_toxic, obscene, threat, insult, identity_hate) in the toxic comment classification dataset under positive label class ‘Toxic’ and work with a binary classification setting. Dataset statistics are present in Appendix A.

### 4.2 Experimental Setting

For identity pairs augmentation, we use GloVe 42B-tokens ³ embedding model [11] with Jaccard-index for computing similarity score between two word sets. We set $r = 0.25$, $topk_1 = 10$, $topk_2 = 500$, and perform lemmatization operation on the generated identity pairs to remove similar form redundant pairs. For Hate Speech Dataset, we keep $n_r = 50\%$, $n_s = 10\%$ of the augmented identity pairs, and $n_b = 10\%$ of the augmented identity terms, for identity replacement, swap, and blindness operations respectively. For Toxic Comment Classification Dataset, we keep $n_r = 10\%$, $n_s = 10\%$, and $n_b = 10\%$.

We work with a set of 10 known identity pairs (shown in Appendix Section B, Table 10), and use the proposed Algorithm 1 to augment identity pairs (shown in Appendix Section B, Table 11).

We conduct experiments with the Convolutional Neural Network (CNN) classifier [9]. Previous studies on Toxic Comment Classification and Hate Speech Classification datasets have demonstrated good performance with the CNN classifier [7], and therefore we use it for experimentation. We use CNN with filters of sizes 3, 4 and 5 with each having 512 filters for all experiments. For the Hate Speech dataset, we train for 25 epochs with early stopping criteria. For the Toxic Comment Classification dataset, we train for 3 epochs. Training batch size across all experiments is 128. We use categorical cross entropy as a loss function. We use Adam with a learning rate $3e^{-4}$ as an optimizer. For CTF metric calculation, we generate all possible combinations of replacements possible with the set of augmented identity pairs.

### 4.3 Results

We use the CTF gap metric proposed by [7] as a fairness evaluation metric where the CTF gap, for an instance, is defined as:

$$
\text{CTF gap}_{\phi_r}(x) = \frac{E_{x_r \sim U[nf]} | f(x) - f(x_r) |}{E_{x_b \sim U[nf]} | \phi_r(x) |} 
$$

CTF gap over test set is the mean of CTF gap over all individual instances (Equation 2) with all possible valid counterfactual instances. We form test set counterfactual instances with augmented identity pairs and IPR operation. Proposed methods could also be tested with other perturbation analysis methods, like Prabhakaran et al. [12], but we limit the paper’s scope to CTF metric.

Table 2 and 3 show performance metric and fairness metric results respectively, for Toxic Comment Classification task. We denote Known Identity Pairs set by ‘KIP’ and Augmented Identity Pairs set by ‘AIP’. We measure Accuracy (at threshold = 0.5) and CTF gap at the label class level (positive class denoted by ‘pos’, negative class denoted by ‘neg’, and combined denoted by ‘all’) to better understand the impact of mitigation methods. Blindness method (CNN AIP Blindness) by design results in a CTF gap value close to 0. Blindness method, however, misses on a lot of information.

---

¹https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge
²https://github.com/Vicomtech/hate-speech-dataset
³https://nlp.stanford.edu/projects/glove/
that may be useful for classification tasks, and hence, is not usually recommended for bias mitigation. Other shortcomings of the blindness method are discussed further in [7]. The augmentation method (CNN AIP Augmentation) adds φφ, φb, and φφ instances for each training instance to the training set and trains the model with the augmented set. CNN AIP Augmentation method, as observed empirically, reduces the CTF gap value. The logit pairing method (CNN AIP LP), as expected, performs best with a significant reduction in the CTF gap value. We present the results with three sets of penalty values, i.e., (1) λ1 = 1, λ2 = 0, (2) λ1 = 0, λ2 = 1, (3) λ1 = 1, λ2 = 1. CNN AIP LP λ1 = 1, λ2 = 1 works better than other methods. Appendix C presents the details about other values of λ1, λ2. We also show with two classifier settings (namely CNN KIP LP and CNN AIP LP) that the proposed identity pairs augmentation method leads to improved CTF gap value. Overall, all the mitigation methods reduce the CTF gap value without harming the classifier’s overall Accuracy and AUC score. We observe similar results for Hate Speech Classification task, as shown in Table 4 and 5, where CNN AIP LP λ1 = 1, λ2 = 1 works best, i.e., shows low CTF gap value and doesn’t affect model’s performance metrics. We visualize AUC vs. CTFall comparison in Figure 2 and 3 to better understand the performance vs. fairness metric trade-off.

5 CONCLUSION

We demonstrate the efficacy of proposed Identity Information Data Augmentation methods, IIDA, to help mitigate bias in text classification tasks. The two-step augmentation setup we propose leads to diverse identity pairs and enhanced training instances based on
simple operations like identity replacement, blindness, and swap. We further show that amalgam of counterfactual and blindness based logit-pairing training scheme results in a more fair classifier. We believe that IIDA’s simple methods could be efficiently leveraged in practice. Continued work on this topic could examine identity pairs validation, perform analysis at different subgroups level, study the effects of out-of-vocabulary tokens, explore perturbation operations enhancement to create and validate meaningful instances.

REFERENCES

[1] Su Lin Blodgett and Brendan O’Connor. 2017. Racial disparity in natural language processing: A case study of social media african-american english. Talk at the 2017 Workshop on Fairness, Accountability, and Transparency in Machine Learning (2017).

[2] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics 5 (2017), 135–146.

[3] Tolga Bolukbasi, Kai-Wei Chang, James Zou, Venkatesh Saligrama, and Adam Kalai. 2016. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. NeurIPS (2016).

[4] Daniel Borkan, Lucas Dixon, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. 2019. Nuanced metrics for measuring unintended bias with real data for text classification. In Companion Proceedings of The 2019 World Wide Web Conference. 491–500.

[5] Ona de Gibert, Naiara Perez, Aitor Garcia-Pablos, and Montse Cuadros. 2018. Hate speech dataset from a white supremacy forum. Proceedings of the 2nd Workshop on Abusive Language Online (ALW2) (2018).

[6] Lucas Dixon, John Li, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. 2018. Measuring and mitigating unintended bias in text classification. In Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society. 67–73.

[7] Sahaj Garg, Vincent Perot, Nicole Limtiaco, Ankur Taly, Ed H Chi, and Alex Beutel. 2019. Counterfactual fairness in text classification through robustness. In Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society. 219–226.

[8] Harini Kannan, Alexey Kurakin, and Ian Goodfellow. 2018. Adversarial logit pairing. CoRR abs/1803.06373 (2018).

[9] Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics, Doha, Qatar, 1746–1751. https://doi.org/10.3115/v1/D14-1181

[10] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. International Conference on Learning Representations (2013).

[11] Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). 1532–1543.

[12] Vinodkumar Prabhakaran, Ben Hutchinson, and Margaret Mitchell. 2019. Perturbation sensitivity analysis to detect unintended model biases. Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP) (2019).

[13] Yusu Qian, Urwa Muaz, Ben Zhang, and Jae Won Hyun. 2019. Reducing gender bias in word-level language models with a gender-equalizing loss function. arXiv preprint arXiv:1905.12601 (2019).

[14] Kellie Webster, Marta Recasens, Vera Axelrod, and Jason Baldridge. 2018. Mind the gap: A balanced corpus of gendered ambiguous pronouns. Transactions of the Association for Computational Linguistics 6 (2018), 605–617.

[15] Jason Wei and Kai Zou. 2019. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP) (2019).
A EXPERIMENT DETAILS

Dataset statistics are present in Table 6 and 7.

| Toxic | Non-toxic | Total |
|-------|-----------|-------|
| Train | 16225     | 143346| 159571|
| Test  | 6243      | 57735 | 63978 |

Table 6: Toxic Comment Classification Dataset

| Hate | No Hate | Total |
|------|---------|-------|
| Train| 957     | 957   | 1914 |
| Test | 239     | 239   | 478  |

Table 7: Hate Speech Dataset

We present results on the Hate Speech Dataset for other configurations of $\lambda_1, \lambda_2$ in Table 8 and 9.

| Model     | Metric | Accuracy | AUC  |
|-----------|--------|----------|------|
| CNN AIP LP $\lambda_1 = 0.5, \lambda_2 = 0.5$ | Acc$_{pos}$ | 0.8243 | 0.7427 | 0.8120 |
| CNN AIP LP $\lambda_1 = 5, \lambda_2 = 5$ | Acc$_{neg}$ | 0.6611 | 0.6025 | 0.6536 |

Table 8: Hate Speech Classification Task: Performance Metric Results

| Model     | Metric | CTF Gap |
|-----------|--------|---------|
| CNN AIP LP $\lambda_1 = 0.5, \lambda_2 = 0.5$ | CTF$_{pos}$ | 0.0151 | 0.0179 |
| CNN AIP LP $\lambda_1 = 5, \lambda_2 = 5$ | CTF$_{neg}$ | 0.0006 | 0.0007 |

Table 9: Hate Speech Classification Task: Fairness Metric Results

B GENERATED IDENTITY PAIRS SET

We here present a small sample of pairs generated by Algorithm 1.

| Term 1  | Term 2          |
|---------|-----------------|
| man     | woman           |
| democrat| republican      |
| trump   | biden           |
| blacks  | whites          |
| asian   | latino          |
| china   | america         |
| africa  | europe          |
| christian| jewish       |
| buddhist| atheist         |
| homosexual| heterosexual |

Table 10: Known Identity Pairs

| Term 1  | Term 2          |
|---------|-----------------|
| female | male            |
| religious | christianity |
| mexican | hispanic       |
| she    | men            |
| japanese | asian        |
| woman  | man            |
| she    | he             |
| muslim | christian      |
| asian  | latino         |
| barack | hillary        |
| gingers | americain  |
| american | woman     |
| liberal | conservative  |
| american | girl        |
| america | him            |
| latino | person         |
| she    | homosexual     |
| homosexual | homosexuality |
| lady   | boy            |
| gay    | lesbian        |
| jew    | christian      |
| atheist| buddhist       |
| asia   | australia      |
| america| japan          |
| latino | african-american|
| she    | person         |
| homosexual | homosexuality |
| lady   | boy            |
| she    | him            |
| africa | zimbabwe      |
| australia | italy    |
| australia | america |
| korea  | europe         |

Table 11: Generated Identity Pairs