Gender-Aware Reinflection using Linguistically Enhanced Neural Models

Bashar Alhafni, Nizar Habash, Houda Bouamor†
Computational Approaches to Modeling Language Lab
New York University Abu Dhabi
†Carnegie Mellon University in Qatar
{alhafni,nizar.habash}@nyu.edu, houamor@qatar.cmu.edu

Abstract

In this paper, we present an approach for sentence-level gender reinflection using linguistically enhanced sequence-to-sequence models. Our system takes an Arabic sentence and a given target gender as input and generates a gender-reinflected sentence based on the target gender. We formulate the problem as a user-aware grammatical error correction task and build an encoder-decoder architecture to jointly model reinflection for both masculine and feminine grammatical genders. We also show that adding linguistic features to our model leads to better reinflection results. The results on a blind test set using our best system show improvements over previous work, with a 3.6% absolute increase in $M^2_F^{0.5}$.

Bias Statement

Most NLP systems are unaware of their users’ preferred grammatical gender. Such systems typically generate a single output for a specific input without considering any user information. Beyond being simply incorrect in many cases, such output patterns create representational harm by propagating social biases and inequalities of the world we live in. While such biases can be traced back to the NLP systems’ training data, balancing and cleaning the training data will not guarantee the correctness of a single output that is arrived at without accounting for user preferences. Our view is that NLP systems should utilize grammatical gender preference information to provide the correct user-aware output, particularly for gender-marking morphologically rich languages. When the grammatical gender preference information is unavailable to the systems, all gender-specific outputs should be generated and properly marked.

We acknowledge that by limiting the choice of gender expression to the grammatical gender choices in Arabic, we exclude other alternatives such as non-binary gender or no-gender expressions. We are not aware of any sociolinguistics published research that discusses such alternatives for Arabic, although there are growing grassroots efforts, e.g., the Ebdal Project.¹

¹https://www.facebook.com/EbdalProject/

1 Introduction

The recent advances in machine learning have propelled the field of Natural Language Processing (NLP) forward at a great pace and raised expectation about the quality of results and especially their impact in a social context, including not only race (Merullo et al., 2019) and politics (Fan et al., 2019), but also gender identities (Font and Costa-jussà, 2019; Dinan et al., 2019; Dinan et al., 2020). Human-generated data, reflective of the gender discrimination and sexist stereotypes perpetrated through language and speaker’s lexical choices, is considered the primary source of these biases (Maass and Arcuri, 1996; Menegatti and Rubini, 2017). However, Habash et al. (2019) pointed out that NLP gender biases do not just exist in human-generated training data, and models built from it; but also stem from gender-blind (i.e., gender-unaware) systems designed to generate a single text output without considering any target gender information. Such systems propagate the biases of the models they use. One example is the I-am-a-doctor/I-am-a-nurse problem in machine translation (MT) systems targeting many morphologically
Table 1: Examples covering all possible combinations of input and output grammatical genders. Changed output words are underlined in the transliterations.

<table>
<thead>
<tr>
<th>Input</th>
<th>Gender</th>
<th>Target Masculine</th>
<th>Target Feminine</th>
</tr>
</thead>
<tbody>
<tr>
<td>Āourney HlwA srych</td>
<td></td>
<td>Āourney HlwA srych</td>
<td>Āourney HlwA srych</td>
</tr>
<tr>
<td>I want quick solutions</td>
<td>B</td>
<td>I want quick solutions</td>
<td>I want quick solutions</td>
</tr>
<tr>
<td>Because I am a blonde woman</td>
<td>F</td>
<td>laṭni ʿarrā ʿashr</td>
<td>laṭni ʿarrā ʿashr</td>
</tr>
<tr>
<td>I am happy [masc.] to meet you</td>
<td>M</td>
<td>ĀnA sycy blqAṣm</td>
<td>ĀnA sycy blqAṣm</td>
</tr>
</tbody>
</table>

Because I am a blonde woman.

I am a [male] doctor’/

AnA mmrDh ‘I am a [female] nurse’, which is inappropriate for female doctors and male nurses, respectively.

In contrast, gender-aware systems should be designed to produce outputs that are as gender-specific as the input information they have access to. Gender information may be contextualized (e.g., the input ‘she is a doctor’), or linguistically provided (e.g., the gender feature provided in the user profile in social media). But, there may be contexts where the gender information is unavailable to the system (e.g., ‘the student is a nurse’). In such cases, generating both gender-specific forms is more appropriate.

In this paper, we present an approach for sentence-level gender reinflection using linguistically enhanced sequence-to-sequence models. Our system takes an Arabic sentence and a given target gender as input and generates a gender-reinflected sentence based on the provided target gender. Table 1 shows some input and output examples. Our work is closely related to the one by Habash et al. (2019), as we use the same corpus that is made available and focus on first-person-singular constructions in Arabic. However, the main contributions of this work are the following: (1) we introduce an approach that jointly models the reinflection for both masculine and feminine grammatical genders, unlike Habash et al. (2019)’s segregated systems; (2) we show that adding linguistic features to our encoder-decoder model leads to better reinflection results. Our code, data, and trained models are publicly available.

This paper is organized as follows. In Section 2, we discuss some related work. In Section 3, we present some Arabic linguistic facts related to grammatical gender. Section 4 introduces our model for joint gender reinflection and describes the encoder-decoder architecture. Then, we present the experimental setup in Section 5 and discuss the results in Section 6. An error analysis is given in Section 7. We conclude and present future work in Section 8.

2 Related Work

Many NLP systems have the ability to embed and amplify societal (gender, racial, religious, etc.) biases across a variety of core tasks such as coreference resolution (Rudinger et al., 2018; Zhao et al., 2018a), machine translation (Rabinovich et al., 2017; Vanmassenhove et al., 2018; Font and Costa-jussà, 2019; Moryossef et al., 2019; Stanovsky et al., 2019; Stafanović et al., 2020; Gonen and Webster, 2020), named entity recognition (Mehrabi et al., 2019), dialogue systems (Dinan et al., 2019), and language modeling (Lu et al., 2018; Bordia and Bowman, 2019).

For the case of gender bias, various research efforts have shown that this could be caused by either human-generated training datasets (Font and Costa-jussà, 2019; Habash et al., 2019), pre-trained word embeddings (Bolukbasi et al., 2016; Zhao et al., 2017; Caliskan et al., 2017; Manzini et al., 2019), or language models (Kurita et al., 2019; Zhao et al., 2019). To mitigate this problem, several researchers

---

2Arabic transliteration is in the HSB scheme (Habash et al., 2007).
3https://github.com/CAMeL-Lab/gender-reinflection
proposed approaches in which they focus mainly on debiasing word embeddings (Bolukbasi et al., 2016; Zhao et al., 2018b; Gonen and Goldberg, 2019) or using counterfactual data augmentation techniques (Lu et al., 2018; Zhao et al., 2018a; Zmigrod et al., 2019; Hall Maudslay et al., 2019).

Most of the solutions were mainly proposed to reduce gender bias in English and may not work as well when it comes to morphologically rich languages. Nevertheless, there have been recent studies that explored the gender bias problem in languages other than English. Zhao et al. (2020) studied gender bias which is exhibited by multilingual embeddings in four languages (English, German, French, and Spanish) and demonstrated that such bias can impact cross-lingual transfer learning tasks. Zmigrod et al. (2019) used a counterfactual data augmentation approach and developed a generative model to convert between masculine and feminine sentences in four languages (French, Hebrew, Italian, and Spanish).

For Arabic, Habash et al. (2019) introduced a two-step approach to gender-identify and reinflect first-person-singular constructions. The identification was done through a feature-based classifier, whereas they used a character-level sequence-to-sequence model for the reinflection. They also compared their two-step approach to a single-step joint identification and reinflection model, which under-performed in the case of the Arabic source (not the machine translation source) task. All of their systems modeled grammatical masculine and feminine genders separately. In this paper, we compare to their results using the publicly available Arabic parallel gender corpus they built – a parallel corpus of first-person-singular Arabic sentences that are gender-annotated and reinflected. However, our work is different from theirs in that we jointly learn reinflection for both masculine and feminine genders together. We also model identification implicitly with reinflection in a single architecture. Furthermore, we formulate the problem as a user-aware grammatical error correction task (UGEC). As such, we use as our primary metric the MaxMatch (M^2) scorer (Dahlmeier and Ng, 2012), which is far more meaningful than the BLEU (Papineni et al., 2002) metric used by Habash et al. (2019) for this task.

3 Arabic Linguistic Background

Modern Standard Arabic (MSA) NLP systems and more specifically those using deep learning, face several challenges when it comes to gender expression including morphological richness, orthographic ambiguity and noise.

Morphological Richness and Complexity  Arabic has a rich morphological system that inflects for gender, number, person, case, state, aspect, mood and voice, in addition to numerous attachable clitics (prepositions, particles, pronouns) (Habash, 2010). This results in a large number of forms for any particular word, with different morpho-syntactic restrictions. For instance, the adjective مَهم mhm̱u ‘important’ [masculine singular indefinite nominative], has a related form مَهمَنَّ mhm̱ṉa that only differs in being accusative in case. In addition to its richness, Arabic morphology has a lot of idiosyncratic inflectional affixes that are not consistent in indicating specific genders or numbers (Alkuhlani and Habash, 2011). For instance, the Ta-Marbuta suffix ذ h, often called the ‘feminine singular ending’, appears with many words where it does not indicate a feminine-singular feature, and cannot be attached to all masculine singular words to turn them feminine. So, in contrast to the good example of مَهمُت mhm̱t ‘important [masculine singular]’, we find words like خَليفة xlyfh ‘Caliph [masculine singular]’, and مَهرة sHrh ‘wizards [masculine plural]’. Furthermore, adding the Ta-Marbuta to some masculine nouns produces nonsensical forms such as رِجلُ رِجل rjl * rjl ‘man-ess (female man)’ from رِجل rjl ‘man’. Similarly, removing the Ta-Marbuta is no guarantee that we map from feminine to masculine in every context. For example, the noun word مَهمَت mhm̱t ‘mission/assignment’ is only feminine and has no meaningful masculine form, as opposed to the adjective مَهمَت mhm̱t ‘important [feminine singular]’ discussed above.

These facts pose major challenges to deep learning models attempting to learn from limited supervised or even large unsupervised data. In this work, we make use of morphological analyzers that indicate all the possible gender information of the words in terms of their functional (grammatical) and form-based (affixational) values (Alkuhlani and Habash, 2011).
Orthographic Ambiguity and Noise  Arabic uses diacritics to specify short vowels and consonantal doubling. These diacritics are optional and generally unwritten, leaving readers to decipher words using contextual and templatic morphology clues. For example, the verb كُنتَ knt can be diacritized as kunta ‘I was’, kunta ‘You [masculine] were’, or kunti ‘You [feminine] were’. This is a challenge for identifying the words that need to change for a first-person target gender. In addition to the issue of orthographic ambiguity, unedited MSA text is reported to be quite noisy with spelling errors reaching $\sim23\%$ of all words (Zaghouani et al., 2014). The most important errors involve Alif-Hamza (Glottal Stop) spelling ($\text{"A}, \text{"A}, \text{"A}$), Ya spelling ($\text{"y}, \text{"y}, \text{"y}, \text{"y}$), and the feminine suffix Ta-Marbuta ($\text{"h}, \text{"h}, \text{"h}$). In Arabic NLP, Alif/Ya normalization is almost standard preprocessing (Habash, 2010). Generally, the high degree of ambiguity and noise result in a high degree of morphological confusability and model sparsity. For instance, a common spelling error of writing the Ta-Marbuta ($\text{"h}$) as Ha ($\text{"h}$) results in interpreting the ($\text{"h}$) as a possessive pronoun clitic attached to a masculine noun: $\text{كَتَبَهُ}$ kAtbh ‘his writer [masculine]’, vs $\text{كَتَبَة}$ kAtb ‘writer [feminine]’.

Normalizing the text may solve some issues related to noise and ambiguity. In this paper, we follow Habash et al. (2019)’s decision to evaluate within an orthographically normalized space for Alif, Ya, and Ta-Marbuta, since the OpenSubtitles 2018 corpus (Lison and Tiedemann, 2016) they use to build the Arabic parallel gender corpus has many of such spelling confusions.

4 Joint Gender Reinflection Model

In this section, we discuss the motivation behind our model architecture as well as the integration of the linguistic features. We also describe the training settings and the model’s hyperparameters for reproducibility.

4.1 Motivation

Sequence-to-sequence models have achieved significant results in grammatical error correction (GEC) (Chollampatt and Ng, 2018; Junczys-Dowmunt et al., 2018; Grundkiewicz et al., 2019) and morphological reinflection tasks (Faruqui et al., 2016; Kann and Schütze, 2016; Aharoni and Goldberg, 2017). Many of these problems are modeled on the word-level, however, such models usually require large amounts of training data to achieve good results. Character-level sequence-to-sequence models can be superior in mitigating the lack of training data and in dealing with subtle morphological reinflection. Further, pre-trained distributed word representations have also shown to be helpful if integrated properly within character-level sequence-to-sequence models (Watson et al., 2018). We formulate the gender reinflection problem as a user-aware grammatical error correction (UGEC) task at the character-level. We also explore leveraging linguistic knowledge on the word-level as well as pre-trained word embeddings to enhance the performance of the model.

4.2 Model Architecture

Given an input sequence $x_{1:n} \in V_x$ containing $k$ words $w_{1:k} \in V_w$, a gender-reinflected output sequence $y_{1:m} \in V_y$, and a target gender $g \in \{F, M\}$, the goal is to model an auto-regressive distribution which is defined over the target vocabulary:\footnote{F stands for Feminine and M stands for Masculine.}

$$P_{V_y}(y_{1:m}|x_{1:n}, g) = \prod_{t=1}^{m} P(y_t|y_{1:t-1}, x_{1:n}, g; \theta);$$

where $\theta$ represents the model’s parameters.

We implement this model using a character-level encoder-decoder neural network with an attention mechanism.
Figure 1: The encoder-decoder architecture for gender reinflection. The input and predicted characters are shown both in Arabic and in the HSB scheme. \(<s>\) and \(</s>\) indicate the start-of-sequence and end-of-sequence tokens respectively. \(\oplus\) refers to the attention mechanism and the filled dot (•) indicates a concatenation operation.

**Encoder** First, each character in the input sequence \(x_i\) is mapped to an embedding \(e_{x_i} \in \mathbb{R}^E\). The character embeddings are parameters of the model which are learned during training. We then feed these embeddings to a two-layer bidirectional GRU (Cho et al., 2014) to obtain a sequence of hidden states \(h^{(e)}_1^t: n\). Each hidden state \(h^{(e)}_i\) is the concatenation of the forward and backward GRU outputs when we feed \(e_{x_i}\).

**Decoder** For the decoder, we use a two-layer GRU with additive attention (Bahdanau et al., 2015; Luong et al., 2015) over the last layer encoder hidden states \(h^{(e)}_1^t: n\). The initial hidden states of the decoder \(h^{(d)}_0\) are learned by passing the encoder hidden states at the last time step \(h^{(e)}_n\) of the corresponding layers through a fully-connected tanh layer, \(h^{(d)}_0 = \text{tanh}(W_h h^{(e)}_n + b_h)\). Given the last layer encoder hidden states \(h^{(e)}_1^t: n\) and the last layer decoder hidden state at the \(t^{th}\) time step \(h^{(d)}_t\), we learn a context vector \(c_t \in \mathbb{R}^{2H}\) that is used to summarize the source attentional context when we predict target symbol \(\hat{y}_t\); we initialize \(c_0 = 0\). At each time step, we feed two inputs to the decoder: the context vector \(c_{t-1} \in \mathbb{R}^{2H}\) and the embedding of the predicted decoder output symbol \(e_{\hat{y}_{t-1}} \in \mathbb{R}^E\) from the previous time step. However, it is important to note that we use scheduled sampling (teacher forcing) (Bengio et al., 2015) with a constant sampling probability during training. The two inputs are then concatenated to create a single vector \(v_t = [e_{\hat{y}_{t-1}}; c_{t-1}] \in \mathbb{R}^{E+2H}\), which is then fed to the GRU to obtain a decoder hidden state \(h^{(d)}_t \in \mathbb{R}^H\). The target gender \(g\) is mapped to an embedding \(e_g \in \mathbb{R}^J\) which is learned during training and concatenated together with the decoder hidden state \(h^{(d)}_t\), the context vector \(c_t\), and the embedding of the predicted symbol from the previous time step \(e_{\hat{y}_{t-1}}\) to create vector \(z_t = [h^{(d)}_t; c_t; e_{\hat{y}_{t-1}}; e_g] \in \mathbb{R}^{H+2H+E+J}\). We finally project \(z_t\) to a vector of size \(|V_y|\) followed by a softmax layer to model the distribution over the target vocabulary \(P_{V_y}(\hat{y}_t) = \text{softmax}(W_y z_t + b_y)\).
Linguistic Features and Word Embeddings We explore adding word-level morphological features as well as pre-trained distributed word representations to the character embeddings. We use the CALIMAStar Arabic morphological analyzer (Taji et al., 2018) to obtain word-level functional gender features (Alkuhlani and Habash, 2011).\footnote{We experimented with both form-based and functional gender features, and found the functional features to be superior in performance; so we only report on them in this paper.} We represent the morphological features for word \( w_j \) as a four-dimension one-hot vector \( \mu_{w_j} \in \mathbb{R}^4 \). Each element of this one-hot vector represents whether the word \( w_j \) is masculine or feminine as well as if the analysis was obtained with or without spelling back-off. We use FastText (Bojanowski et al., 2017) to learn distributed word representations and we denote the FastText word embedding for word \( w_j \) as \( \rho_{w_j} \in \mathbb{R}^F \).

Similarly to Watson et al. (2018), we added the word-level features to the character embeddings only on the encoder side. Each character embedding \( e_{x_i} \) is then enriched with \( \rho_{w_j} \) and \( \mu_{w_j} \) to create a single vector \([e_{x_i}; \mu_{w_j}; \rho_{w_j}] \in \mathbb{R}^{E+4+F}\) which we feed to the encoder, where \( w_j \) is the word containing character \( x_i \).

Inference At inference time, we use greedy decoding to find the most likely sequence:\footnote{It is important to note that we also explored beam search for decoding, however, greedy decoding yield better results.}

\[
\hat{y}_{1:m} = \arg\max_{\hat{y} \in V_y} P(\hat{y} | x_{1:n}, g) = \arg\max_{\hat{y} \in V_y} \prod_{t} P(\hat{y}_{1:t-1}, x_{1:n}, g)
\]

The architecture of our gender reinflection linguistically enhanced sequence-to-sequence model is shown in Figure 1.

4.3 Training Settings

For all the experiments described in this paper, we use a batch size of 32, a character embedding size of \( E = 128 \), a gender embedding size of \( J = 10 \), a hidden size of \( H = 256 \), a scheduled sampling probability of 0.3, a dropout probability of 0.2, and gradient clipping with a maximum norm of 1. The FastText embeddings have a dimension of \( F = 100 \) and were trained for 10 epochs using the OpenSubtitles 2018 corpus in a skip-gram manner with context windows of 2 and 3 respectively. We train the model for 50 epochs by minimizing the average cross-entropy loss defined as follows:

\[
\mathcal{L}(y_{1:m}, \hat{y}_{1:m}; \theta) = \frac{1}{m} \sum_{t=1}^{m} \mathcal{L}(y_t, \hat{y}_t; \theta); \mathcal{L}(y_t, \hat{y}_t; \theta) = -\log P_{V_y}(\hat{y}_t)
\]

We use the Adam optimizer (Kingma and Ba, 2014) with an initial learning rate of 0.0005, decaying by a factor of 0.5 if the loss on the development set does not decrease after 2 epochs.

5 Experiments and Evaluation

In this section, we discuss the data we use to train and evaluate our models. We also discuss the evaluation metrics and the various systems we implemented including the baselines.

5.1 Data

For our experiments, we use the publicly available Arabic parallel gender corpus (Habash et al., 2019), containing 12,238 parallel gender-annotated sentences: F (feminine), M (masculine) or B (gender-ambiguous). The corpus is divided into three parallel balanced corpora: (1) CorpusIn, containing F, M and B sentences, (2) CorpusM containing M and B sentences only, and (3) CorpusF containing F and B sentences only.\footnote{In this work, we consider the B cases to be masculine in CorpusM and feminine in CorpusF.} Table 1 shows examples of what CorpusIn (Input), CorpusM (Target Masculine), and CorpusF (Target Feminine) would look like.

We build our target corpus by concatenating CorpusM and CorpusF, while our source corpus is a duplication of CorpusIn. Since our goal is to build a single user-aware joint gender reinflection model
for both grammatical genders, we introduce the notion of target gender $g$ having two possible values: F or M. All of the target sentences from CorpusM will have an M target gender, whereas all of the target sentences from CorpusF will have an F target gender. We follow the same data split as Habash et al. (2019). After merging the corpora we ended up with 17,132 sentence pairs for training (TRAIN), 2,448 for development (DEV), and 4,896 for testing (TEST). All of our systems are trained to take a source sentence and a target gender as input to produce a gender-reinflected target sentence as described in section 4.2.

5.2 Metrics
Gender Reinflection We follow Habash et al. (2019) and use BLEU as an evaluation metric (Papineni et al., 2002), however, we believe that BLEU is not a suitable metric for our task due to the high similarity between the input and output sentences. We use SacreBLEU (Post, 2018) to compute the BLEU scores. Additionally, we use the MaxMatch (M²) scorer (Dahlmeier and Ng, 2012) to compute the word-level edits between the input and reininflected output. We report the precision, recall, and $F_{0.5}$ scores calculated against the gold edits, which were also created by the M² scorer. We are aware that there are other tools to consider for word-level edit calculation such as ERRANT (Bryant et al., 2017), but we did not use them as they require additional dependencies to work for Arabic.

Input Gender Identification Our sequence-to-sequence model does not explicitly identify the gender of the input sentence; however, we consider any attempted change (or lack thereof) to the input as a signal for the implicit gender identification: if our model reinflects the source sentence, then we consider the gender of this sentence to be the opposite of the given target gender. But if the model does not reinflect the source sentence, then we consider the gender of this sentence to be the same as the target gender. We report the average $F_1$ score for M and F gender identification over the source sentences.

We report the results for gender identification and reinflection in a normalized space for Alif, Ya, and Ta-Marbuta as discussed in section 3.

5.3 Baselines
In addition to comparing with the results from Habash et al. (2019), we include two baselines. The first one is a DO NOTHING baseline which simply passes the input to the output as is. This baseline is intended to show how similar the inputs and the outputs are. The second is a baseline in which we define a bigram maximum likelihood estimation (MLE) model: given an input sequence of words $x_{w1:n} \in V_x$, a target sequence of words $y_{w1:n} \in V_y$, and a target gender $g \in \{F, M\}$, the MLE model is built as follows:

$$P(y_{w1:n} | x_{w1:n}, x_{w1:n-1}, g) = \frac{\text{count}(y_{w1:n}, x_{w1:n}, x_{w1:n-1}, g)}{\text{count}(x_{w1:n}, x_{w1:n-1}, g)}$$

At inference time, we pick the target word $\hat{y}_{w1:n}$ which maximizes the probability defined above. If $\hat{y}_{w1:n}$ was not observed in the training data along with $x_{w1:n}$ and $x_{w1:n-1}$, we back-off to a lower-order distribution (unigram) $P(\hat{y}_{w1:n} | x_{w1:n}, g)$. In the worst case scenario, where $\hat{y}_{w1:n}$ was not observed in the training data along with $x_{w1:n}$, we pass $x_{w1:n}$ to the output.

The MLE baseline is suitable for our case because the input and output sentences are perfectly aligned on the word-level.

5.4 Systems
We explore four variants of the model described in section 4.2. In the first, we provide the encoder with the character embeddings without any morphological features or FastText embeddings and we refer to it as JON. The second variant is where we add the morphological features to the character embeddings but without the FastText embeddings and we refer to it as JON+MORPH. For the third variant, we explore adding both the morphological features and the FastText embeddings to the character embeddings, we refer to it as JON+MORPH+FT. To build the fourth one, we selected the best variant and trained it in a similar fashion to Habash et al. (2019). We trained two systems disjointly; one using CorpusM and the

---

8We experimented with different n-gram sizes for the MLE model, the bigram yielded the best results.
Table 2: Results of a number of systems on the DEV set.

<table>
<thead>
<tr>
<th>Reinflection</th>
<th>Identification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precision</td>
<td>Recall</td>
</tr>
<tr>
<td>DO NOTHING</td>
<td>100.0</td>
</tr>
<tr>
<td>MLE (bigram)</td>
<td>65.5</td>
</tr>
<tr>
<td>Habash et al. (2019)</td>
<td>74.0</td>
</tr>
<tr>
<td>JOINT</td>
<td>70.6</td>
</tr>
<tr>
<td>JOINT+MORPH</td>
<td>75.3</td>
</tr>
<tr>
<td>JOINT+MORPH+FT</td>
<td>64.8</td>
</tr>
<tr>
<td>DISJOINT+MORPH</td>
<td>63.6</td>
</tr>
</tbody>
</table>

Table 3: Results of baseline systems and the best system on the TEST set.

<table>
<thead>
<tr>
<th>Reinflection</th>
<th>Identification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precision</td>
<td>Recall</td>
</tr>
<tr>
<td>DO NOTHING</td>
<td>100.0</td>
</tr>
<tr>
<td>MLE (bigram)</td>
<td>70.8</td>
</tr>
<tr>
<td>Habash et al. (2019)</td>
<td>77.7</td>
</tr>
<tr>
<td>JOINT+MORPH</td>
<td>79.0</td>
</tr>
</tbody>
</table>

other using Corpus5 and reported the average performance of both systems. We refer to this last variant as DISJOINT+MORPH.

6 Results

The results of our evaluation on the DEV set are presented in Table 2. The best performing system is JOINT+MORPH. It improves over the previous SOTA on this task, Habash et al. (2019), in every compared metric, including a 4.4% absolute increase in M² F₀.₅. The biggest contribution to the performance increase is from recall (10.3% absolute). In fact, all of the neural models we introduced in this paper improve over the Habash et al. (2019) results in terms of recall (at varying degrees); however, only JOINT+MORPH improves in terms of recall and precision. The MLE results are surprisingly competitive in terms of precision, scoring higher than some of the weaker neural models; while being the worst (barring DO NOTHING) across all other metrics.

The two aspects of our best system (being joint and using morphological features) are important to its performance. When we compare JOINT+MORPH to its JOINT counterpart, we observe an 5.6% absolute increase in the M² F₀.₅ score and a corresponding 0.6% increase in identification F₁ score. This confirms that morphological features are helpful for both gender identification and reinflection.

An ablation experiment comparing the best system JOINT+MORPH to the disjoint variant of it (DISJOINT+MORPH) demonstrates the large added value of using a joint model: an 11.2% absolute increase in M² F₀.₅ score, 0.45 BLEU points, and 0.8% absolute improvement in identification F₁ score. The use of word embeddings was not helpful to our best system. One possible explanation is that the use of semantically oriented embeddings may not be optimal for fine-targeted rewriting tasks.

The results on the TEST set using the baselines and the best system from the DEV experiments are given in Table 3. These results show consistent conclusions with the DEV results. Our best system improves over the previous SOTA in every compared metric, including a 3.6% absolute increase in terms of M² F₀.₅.
Table 4: Summary of the errors found in the Dev set organized by target gender (M or F) and in combination (M+F).

<table>
<thead>
<tr>
<th></th>
<th>M Target</th>
<th>F Target</th>
<th>M+F Target</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Change</td>
<td>35 64%</td>
<td>52 71%</td>
<td>87 68%</td>
</tr>
<tr>
<td>Wrong Change</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Case form</td>
<td>17 31%</td>
<td>14 19%</td>
<td>31 24%</td>
</tr>
<tr>
<td>Uninfectable word</td>
<td>9 16%</td>
<td>0 0%</td>
<td>9 7%</td>
</tr>
<tr>
<td>Odd characters</td>
<td>2 4%</td>
<td>6 8%</td>
<td>8 6%</td>
</tr>
<tr>
<td>Other</td>
<td>2 4%</td>
<td>3 4%</td>
<td>5 4%</td>
</tr>
<tr>
<td>Gold Error</td>
<td>3 5%</td>
<td>7 10%</td>
<td>10 8%</td>
</tr>
<tr>
<td>Total</td>
<td>55 100%</td>
<td>73 100%</td>
<td>128 100%</td>
</tr>
</tbody>
</table>

7 Error Analysis

We conducted a manual error analysis examining all of the errors in the output of our best system on the Dev set. In total, there were 106 sentences with errors (or 4.3% out of 2,448). In those erroneous sentences, there were 128 words with problems. Table 4 presents the detailed scores, which we discuss next.

Around two thirds of the word errors were false negatives, i.e., where a change should have happened but did not (Table 4 No Change). In a quarter of the No Change cases, a clear copular construction context for first person gendered expression is seen. For example, the word \( fnAn \) ‘artist [masc]’ in \( \text{\'artist [fem]'} \) is not correctly reinflected to its F target form \( fnAnh \). The No Change errors with target gender F are 50% higher than the target gender M; this suggests that the system is more adept at identifying feminine source text than the other way around. This is plausible given that the Arabic feminine form is the marked variety.

Returning to the rest of the errors, an additional quarter of them involved a false positive (Table 4 Wrong Change). Three types of incorrect changes are noteworthy. First is imperfectly reinflecting the masculine form by failing to indicate case (Table 4 Case form), e.g., generating \( knt m\ddot{s}\gamma l \) instead of \( knt m\ddot{s}\gamma lA \) ‘I was busy [masc]’. It should be noted that such cases are commonly used and are ‘accepted’ since most modern dialects of Arabic lost the productive generation of case. Second is reinflecting words that are not inflectable for gender (Table 4 Uninfectable word). One example is adding the feminine nominal suffix \( \ddot{h} \) to the first person imperfective verb \( \dddot{\text{Am\ddot{b}l}} \) in \( \dddot{\text{Am\ddot{b}l j\dot{s}c Al\ddot{sr}kAt}} \) ‘I represent corporate greed’. This results in creating a nonsensical verbal form \( \dddot{\text{Am\ddot{b}l}} \dddot{h} \) which is a homograph with the word ‘examples’. The third type of change errors involves random generation of odd repetitive character sequences (Table 4 Odd characters), a side effect of using character sequence-to-sequence models. One example in our data is the generation of the nonsensical form \( qqq \) from the word \( q\ddot{l}q \) ‘worried [masc]’ instead of \( q\ddot{l}q\ddot{h} \) ‘worried [fem]’. Finally, about 1/12\textsuperscript{th} of all counted errors are miscounts due to Gold annotation fails, where our system actually generated the correct output (Table 4 Gold Error).

Considering the detailed scores for the whole DEV set and for M target and F target cases, we note the following. As expected, the F target setting has more errors than the M target setting. No Change errors and Gold errors are more common for the F target setting. The Case form errors are only seen in the M target setting. Errors with uninflectable words are almost equally present. These errors suggest that more work needs to be done on identifying when a reinflection should take place. Furthermore, to address the errors of uninflectable forms and case-marked forms, we may have to incorporate more linguistic knowledge or more powerful language models.
8 Conclusion and Future Work

In this paper, we proposed a solution to single-output NLP systems that allows users to specify their grammatical gender preference in Arabic. Our intention is to enable users to reduce the harm that may be produced by NLP systems propagation of biased representations. Our joint approach for sentence-level gender reinflection uses linguistically enhanced sequence-to-sequence models and frames the problem as a user-aware grammatical error correction task. Our system takes an Arabic sentence and a given target gender as input and generates a gender-reinflected sentence based on the provided target gender. We showed that linguistic knowledge helps in learning gender identification implicitly which improves reinflection results. In future work, we would like to explore different architectures such as Transformer-based models (Vaswani et al., 2017). Furthermore, we are interested in exploring the added value of combining syntactic and morphological features. We would also like to apply our approach to different languages and dialectal varieties. Lastly, we plan to extend the Arabic parallel gender corpus beyond first-person-singular constructions and adapt our models accordingly.

Acknowledgements

This research was carried out on the High Performance Computing resources at New York University Abu Dhabi (NYUAD). We would like to thank the Computational Approaches to Modeling Language Lab at NYUAD for their help and invaluable suggestions throughout this project. We also would like to thank Professor Dima Ayoub for helpful conversations.

References


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.


Emily Dinan, Angela Fan, Adina Williams, Jack Urbanek, Douwe Kiela, and Jason Weston. 2019. Queens are powerful too: Mitigating gender bias in dialogue generation. *ArXiv*, abs/1911.03842.


Hila Gonen and Yoav Goldberg. 2019. Lipstick on a pig: Debiasing methods cover up systematic gender biases in word embeddings but do not remove them.


Thomas Manzini, Yao Chong Lim, Yulia Tsvetkov, and Alan W Black. 2019. Black is to criminal as caucasian is to police: Detecting and removing multiclass bias in word embeddings.

Ninareh Mehrabi, Thamme Gowda, Fred Morstatter, Nanyun Peng, and Aram Galstyan. 2019. Man is to person as woman is to location: Measuring gender bias in named entity recognition.


Jieyu Zhao, Subhabrata Mukherjee, Saghar Hosseini, Kai-Wei Chang, and Ahmed Hassan Awadallah. 2020. Gender bias in multilingual embeddings and cross-lingual transfer.