Introducing new Gender Signals to Language Models

Faezeh Hosseini
faxhosseini@gamail.com

Ali Sarmadi
sarmadiali98@gmail.com

Abstract

Biases in language models can lead to unfair or inaccurate outcomes in natural language processing. Some techniques try to remove these biases by applying post-hoc debiasing methods, such as Zari, a debiased language model based on BERT large. However, these methods may also affect the model's ability to deal with gender-specific expressions in context, which can limit the model's accuracy and usefulness. In this project, we extract gender-specific knowlege from Wikidata (Vrandečić and Krötzsch, 2014), and then use a prompt-based method to add that knowledge to the de-biased language model, without bringing back the biases that were eliminated. We test our method on different datasets that measure the model's gender bias and gender sensitivity, and show that our method increases the model's gender-specific score, while, in some models, keeping its low gender bias score. We argue that our method can improve the gender sensitivity of de-biased language models.

1 Introduction

Language models are powerful tools for natural language processing, but they often suffer from biases that reflect the data they are trained on. For example, some language models may associate certain occupations or attributes with specific genders, which can lead to unfair or inaccurate outcomes. To address this problem, many techniques have been proposed to mitigate the biases in language models, such as data augmentation, adversarial training, or post-hoc debiasing. However, these techniques may also have some drawbacks, such as reducing the model's performance on other tasks or losing the model's ability to handle gender-specific expressions in context. In this project, we focus on a debiased language model named Zari, which is based on BERT (Devlin et al., 2019) large and uses a post-hoc debiasing method. Zari was evaluated on five datasets (T1-T5) that measure the model's gender bias and gender sensitivity, as described in the paper "DIFAIR: A Benchmark for Disentangled Assessment of Gender Knowledge and Bias" (Zakizadeh et al., 2023). The results showed that Zari achieved a low gender bias score, indicating that the creators were successful in removing the unwanted biases from the original BERT model. However, Zari also had a low gender-specific score, meaning that it failed to recognize the genders correctly when they were relevant or important for the context. For example, Zari might not be able to distinguish between "he" and "she" pronouns or generate appropriate gendered words in a sentence completion task. This suggests that Zari lost some of its gender-specific knowledge during the debiasing process, which could limit its usefulness for some applications. Therefore, in this work, we propose a method to inject gender-specific knowledge to the de-biased language model, without reintroducing the biases that were removed. Our method aims to enhance Zari's gender sensitivity and improve its gender-specific score, while preserving its low gender bias score.

2 Literature Review

Masked Language Models (MLMs) are powerful pre-trained models that can generate text representations for various downstream natural language processing (NLP) tasks. However, MLMs have been shown to exhibit social biases, such as gender and race biases, that can negatively affect the performance and fairness of the NLP systems that use them. Therefore, it is important to evaluate and mitigate the social biases in MLMs before deploying them in real-world applications.

Several studies have proposed methods to debias MLMs by modifying their pre-training objectives, data, or parameters. For example, (Bolukbasi et al., 2016) proposed a method to debias word embeddings by removing the gender subspace from the embedding space. (Zhao et al., 2018) pro-

posed a method to debias sentence encoders by adding an adversarial component that prevents the encoder from predicting the gender of the input sentence. Kaneko and (Kaneko et al., 2022) proposed a method to debias MLMs by masking the gendered words during pre-training.

However, debiasing MLMs is not enough to ensure their social fairness in downstream tasks. (Kaneko et al., 2022) conducted a comprehensive study on the relationship between task-agnostic intrinsic and task-specific extrinsic social bias evaluation measures for MLMs, and found that there exists only a weak correlation between these two types of evaluation measures. Moreover, they found that MLMs debiased using different methods still re-learn social biases during fine-tuning on downstream tasks. They identified the social biases in both training instances as well as their assigned labels as reasons for the discrepancy between intrinsic and extrinsic bias evaluation measurements.

One possible way to address this issue is to add gender-specific information to debiased MLMs in a controlled manner, so that they can preserve factual gender information without reinforcing harmful stereotypes. For example, (Liu, 2021) proposed a method to inject world knowledge into language models through soft prompts, and demonstrated its effectiveness on several knowledge-intensive tasks.

Zari is a collection of dataset and models for gendered ambiguous pronoun resolution, which is a subtask of coreference resolution. It was created by Google Research and released on GitHub. Zari contains 2,000 sentences from Wikipedia with ambiguous pronouns, such as he or she, that refer to one of two candidate names in the same sentence. The sentences are balanced by gender and genre, and annotated by human raters with the correct referent for each pronoun. Zari also provides several models based on BERT that achieve high accuracy and low gender bias on the dataset.

Zari is an extension of the GAP dataset , which was the first gender-balanced corpus of ambiguous pronouns. GAP was designed to evaluate and mitigate the gender bias in coreference resolution systems, which tend to favor masculine entities over feminine ones. GAP contains 8,908 sentences from Wikipedia with ambiguous pronouns, and has been used as a benchmark for several studies and competitions on pronoun resolution (Ionita et al., 2019; Larin et al., 2019; Webster et al., 2019).

Zari differs from GAP in several aspects. First, Zari has a larger number of candidate names per sentence (2 vs 1), which makes the task more challenging and realistic. Second, Zari has a more diverse set of genres, such as sports, arts, politics, and science, which can introduce different types of biases and linguistic phenomena. Third, Zari has a more fine-grained annotation scheme, which distinguishes between cases where the pronoun refers to one of the candidates, none of the candidates, or both of the candidates (in case of plural pronouns). Fourth, Zari provides pretrained models that can be easily used or adapted for pronoun resolution tasks.

For measuring the gender-specific socres of our resulted models, we have used "DIFAIR: A Benchmark for Disentangled Assessment of Gender Knowledge and Bias", which is a dataset that evaluates language models on their gender bias and gender knowledge using a masked language modeling objective. The authores examine various models and debiasing techniques on this dataset, and they observe that they all exhibit challenges with gender. They either fail to differentiate between genders when appropriate or they compromise gender facts when they attempt to achieve fairness. They also propose a new metric that quantifies both fairness and performance on gendered instances.

In this project, we aim to explore the effectiveness of adding gender-specific information to Zari. We hypothesize that this approach can enhance the factual accuracy and social fairness of the generated text representations, while relatively and in select models, avoiding the re-learning of social biases.

3 Methodology

We have taken three steps in order to complete this project:

1. Creating a Gender-Specific dataset.
2. Injecting the Gender-Specific knowledge into an already debiased model.
3. Measuring the Gender-Specific score and Gender bias in the resulted model.

3.1 Creating a Gender-Specific Dataset

As previously mentioned, we adopt knowledge base (KB) triples from Wikidata, which is a free, open, and trustworthy knowledge base that anyone can use. It is a project of the Wikimedia Foundation, the same organization that runs Wikipedia

and other online encyclopedias. Wikidata aims to provide a common source of structured data for all Wikimedia projects, as well as for other applications and services on the web.

Wikidata stores data in the form of items, which are entities that have a unique identifier and a label. Each item can have multiple statements, which are facts or claims about the item. Each statement consists of a property and a value, which can be another item, a literal, or a complex data type.

We have filtered the triples of Wikidata using 191 one-token gender-specific words such as: 'actors','actresses','airman', 'airwoman', 'airmen', 'airwomen', 'aunts', 'uncles', 'boys', 'girls', 'brides', 'grooms', etc. These words have one thing in common: They all contain gender-specific information as well as extra information about a person. This filtering resulted in a dataset with 19,498 triples that contains gender-specific knowledge.

A complete list of these gender-specific words alogside with a histogram distribution of the most repeated ones can be seen in the appendix.

3.2 Injecting the Gender-Specific knowledge into an already debiased model

Soft prompts are learnable embeddings that are added to the input of the LM, and act as a task-specific guide for the model. Soft prompts can be seen as a way of finetuning with fewer parameters, as they only need updating a small number of embeddings, while keeping the rest of the LM parameters fixed. Soft prompts can also be seen as a way of priming for specific tasks, as they give a contextual hint for the model to produce the right output for the task.

In this work, we want to train soft knowledge prompts (KPs) to store gender-specific knowledge, which could work as an external memory for LMs. We think that KPs can help LMs keep factual gender information without making harmful stereotypes worse. We use gender-specific KB triples from Wikidata (Vrandečić and Krötzsch, 2014), as a simple and reliable source of world knowledge. We train KPs with a masked language modeling (MLM) objective, where the goal is to produce the relation between subject and entity of a KB triple given the subject entity and object.

3.2.1 Soft Prompts

Finetuning language models can be costly and time-consuming, as it requires updating millions of parameters and storing multiple versions of the same model. Moreover, finetuning LMs can also lead to overfitting or catastrophic forgetting, which means that the model may lose its generalization ability or forget some of the original knowledge that was useful for other tasks.

To address these challenges, some researchers have proposed to use soft prompts instead of finetuning LMs. Soft prompts can be seen as a form of parameter-efficient finetuning, as they only require updating a small number of embeddings, while keeping the rest of the LM parameters frozen. Soft prompts can also be seen as a form of task-specific priming, as they provide a contextual cue for the model to generate the appropriate output for the task.

Different approaches have been recently proposed to train soft prompts (Lester et al., 2021; Li and Liang, 2021; Hambardzumyan et al., 2021).

One of the most popular methods (Lester et al., 2021), and probably the simplest one, consists of the following steps:

1. For a task in the dataset, prepend a fixed number of embeddings (soft prompts) to the word embeddings of every input.

2. During finetuning, update the soft-prompt while keeping all the other parameters of the LM frozen.

Despite its simplicity, this method has been shown to achieve competitive results on various natural language understanding and generation tasks, such as natural language inference, sentiment analysis, or text summarization.

3.2.2 Soft Knowledge Prompts

We are interested in training soft knowledge prompts (KPs) to encode gender-specific knowledge, which could work as an external memory for LMs. In this work, we focus on the training of entity-centric KPs, each of which stores the knowledge related to a specific entity from a knowledge base (KB). In other words, the KP of an entity encodes information from the KB triples that mention the entity either as a subject or an object. We adopt gender-specific KB triples from Wikidata (Vrandečić and Krötzsch, 2014), as a simple and trustworthy source of world knowledge.

3.2.3 KP Models

Knowledge prompts are learnable embeddings that are prepended to the input of the NLP model, and

act as an external memory for the model. Knowledge prompts can store and retrieve information or knowledge from various sources, such as knowledge bases, ontologies, or databases. Knowledge prompts can enhance the factual accuracy and social fairness of the output of the NLP model, while avoiding the re-learning of biases or errors during finetuning.

Knowledge prompts can be either entity-centric or relation-centric. Entity-centric knowledge prompts store the knowledge related to a specific entity from a knowledge base (KB). For example, an entity-centric knowledge prompt for Barack Obama could encode information from the KB triples that mention him either as a subject or an object. Relation-centric knowledge prompts store the knowledge related to a specific relation from a KB. For example, a relation-centric knowledge prompt for spouse could encode information from the KB triples that mention this relation.

Knowledge prompts can be trained with different objectives, such as masked language modeling (MLM), generative modeling, or classification modeling. MLM is an objective where the goal is to predict a masked word or entity in a given text or KB triple. Generative modeling is an objective where the goal is to generate a natural language sentence or paragraph given a text or KB triple.

Knowledge prompts can be used for various natural language understanding and generation tasks, such as natural language inference, sentiment analysis, text summarization, machine translation, question answering, or dialogue generation. Knowledge prompts can improve the performance and fairness of NLP models on these tasks, while avoiding the re-learning of biases or errors during finetuning.

- **Using Neural Networks as Knowledge Prompts in the Tokenization Phase**

This was the initial idea that occurred to us for implementing the Knowledge Prompt model. We intended to use a simple neural network as KP and train it by passing the tokenized input and obtaining the output of this network as a soft prompt. Then we intended to combine this soft prompt with the tokenized input in order to inject its knowledge and input the result to the language model Zari. Then we could update this network with the losses in the training procedure. But it transpired that the output of the neural network could not be combined with the input because we were actually combining two sequences that were tokenized and they should consist of valid token ids. But what we obtained as soft prompt from the neural network output might not have valid token ids, so we encountered the error that indicated the input to the Language Model was not valid in the vocabulary of the model. In other words, we attempted to combine tokenized data but what we obtained from the neural network did not have valid token ids in the language model's vocabulary, so this idea was a failure.

For instance, suppose we have a KB triple like "Barack Obama | husband | Michelle Obama" in the first place. We would tokenize the triple using Zari's tokenizer, and obtain something like:
`[101, 1441, 8112, 1024, 8776, 1024, 4718, 8112, 102] .`
Then we would mask one of the entities, say Michelle Obama, and end up with something like:
`[101, 1441, 8112, 1024, 8776, 1024, [MASK], [MASK], 102].`
Then we would pass this sequence to a simple neural network as KP, and get an output like:
`[0.1, 0.2, ..., 0.9].`
Then we would try to combine this output with the original input sequence, and obtain something like:
`[101, 1441, 8112, 1024, 8776, 1024, [MASK], [MASK], 102, 0.1, 0.2, ..., 0.9].`
But this sequence is not valid for Zari's vocabulary, because it contains numbers that are not token ids. So Zari would throw an error and reject this sequence.

This is why our idea did not work. We realized that we need a different method to combine the output of the neural network with the input of the language model. We also realized that we need a different type of neural network as KP, one that can produce valid token ids or embeddings that can be merged with the input embeddings.

- **Using Neural Networks as Knowledge Prompts in the Embedding Layer**

After the previous failed approach, we searched for another way to combine the input and the soft prompt, which has the gender-specific knowledge. So we used the embedding layer of the language model Zari.Suppose we have a KB triple like "Vanessa Kerry | father | John Kerry". Our objective here is to mask the relation that most of the time is gender-specific, so we will have "Vanessa Kerry [MASK] John Kerry" and save the relation

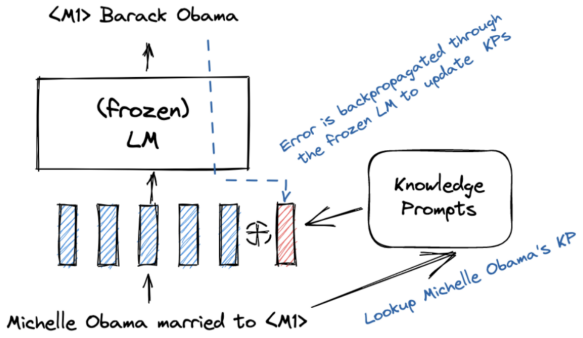

Figure 1: Injecting knowledge to LM through soft prompts

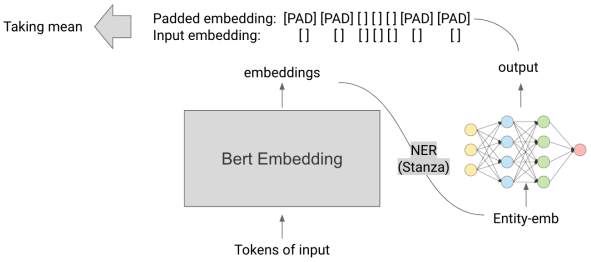

Figure 2: getting the knowledge prompt form KP which is a Neural Network and taking the mean of obtained soft embedding and input embedding. The result is the input to LM.

as the label, so we can calculate the loss in the feature and backward it in order to update our network. We want to use this objective to train a knowledge prompt model based on recognized entities in triples.

As an example, we use the example of the paper (dos Santos et al., 2022) that is shown in Figure 1, but the structure of the Kp that was proposed in this article wasn't proper in our scale. The proposed KP structure in Figure 1 is a table, like a database where each entity has its own row of knowledge and you can get the knowledge of an entity by passing it to this table. But in our work, because our data is limited we choose another structure for our KP which is the Neural Network structure. And also we have a different objective compare to this example. This way, we can inject gender-specific knowledge into Zari's input embeddings without changing its parameters or vocabulary.

As we mentioned before, in this approach we use the embedding layer of the model(Zari) and will add a Neural Network module in this layer in order to learn the embedding of gender-specific words like entities here. Until this step we made the masked-triple, we will pass it to the model Zari that has an added module Soft-Embedding to its embedding layer. In this module we input the masked-triple to the model Stanza for the task Name Entity Recognition, we get the person entities of the input by model Stanza(if there was any entity in the masked-triple). Then we want to add the gender knowledge of this entity to the input(masked-triple), so we pass this entity embedding to the Neural Network module and get the soft-embedding out of it. After this step we will take the mean of these two embedding vectors, which are the masked-triple embedding and the entity-embedding vectors. Here, we obtain the leaned-embedding vector that is the input embedding for the subsequent layers of the model. We do calculating the mean of two vectors, with respect to position alignment of recognized entities. You can see this levels in Figure 2

We will get the predicted mask out of the LM Zari and calculate the loss and backward it in order to update the Neural Network weights.

As an example, we use the example of the paper (soft prompt paper) that is shown in Figure 1, but the structure of the Kp that was proposed in this article wasn't proper in our scale. The proposed KP structure in Figure 1 is a table, like a database where each entity has its own row of knowledge and you can get the knowledge of an entity by passing it to this table. But in our work, because our data is limited we choose another structure for our KP which is the Neural Network structure. This way, we can inject gender-specific knowledge into Zari's input embeddings without changing its parameters or vocabulary.

- **Using a Simple Neural Network as Knowledge Prompts**

For this section, we used a simple neural network as a knowledge prompt for training on gender-specific knowledge. This simple neural network has just two layers: input and output. The input layer has a size of (BertEmbedding.size , 100) and the output layer has a size of (100 , BertEmbedding.size). The results of training this model on our dataset and the evaluation scores are shown in table 1.

- **Using a More Complex Neural Network as Knowledge Prompts**

This approach is like the previous one the only thing that is different is the structure of NN KP, and

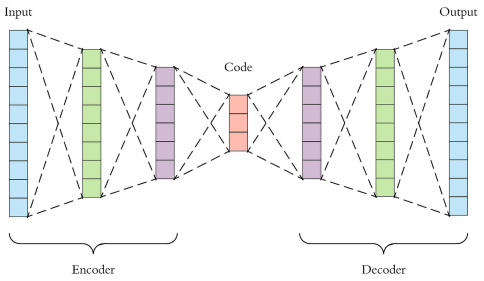

Figure 3: autoencoder model architecture

it's a bit more complex. The structure of this Neural Network is as follows: This model is consisting of 3 layers: input, output, and one hidden layer. The input layer has a size of (BertEmbedding.size, 100), the hidden layer has a size (100,1000) and the output layer has a size of (1000, BertEmbedding.size).

- **Using an Auto-Encoder as Knowledge Prompts**

In this part we used a Neural Network that has an autoencoder architecture. You can see this architecture in Figure 3. We believe that a more complex NN model leads to better and more accurate results and we did get better results and higher bias score of this model. Although we trained this network in 100 epochs on our dataset and we achieved the results that you can see in table for the model3 that is the information of performance of our autoencoder model.Also you can see the training results after 80 epochs in Figures 8,9.

3.3 Measuring the Gender-Specific knowledge and Gender bias in the resulted model

For measuring gender-specific knowledge and gender bias, we have used "DIFAIR: A Benchmark for Disentangled Assessment of Gender Knowledge and Bias", a dataset that tests how well language models can handle gender bias and gender facts at the same time. The authors use a masked language modeling task to acomplish this task. They compare different models and debiasing methods on this dataset, and they find that they all have problems with gender. They either don't know when to use gender or they lose gender facts when they try to be fair. They also make a new metric that measures both fairness and performance on gendered sentences. Their gender-specific dataset is consisted of 5 sections: T1 - T5. Each of which measure a different aspect of gender-specific knowledge in a model The higher score our model achieves in each of these sections, the more knowledge it has. Their gender-neutral datset determines whether our model performs impartially in gender-neutral situations. Which translates into weather our model is biased or not. The higher score we get, the less biased is our model. Finally, the total bias score is defined as an average of the T1-T5 score and the bias score, which means naturaly, the higher it is, the better our model has performed in total.

4 Results

The results of our project are presented here in two parts. In the first part, the training loss and accuracy resutls for different models are demonstrated. Also, in the second section, the different types of bias scores for different approaches and models can be seen.

4.1 Knowledge Prompt Training Results

We train KPs with a masked language modeling (MLM) objective (Devlin et al., 2019; Taylor,1953), where the goal is to generate the relation between the subject and the object of a KB triple given the subject entity and the object entity.

- **Training Results for Simple Neural Network as a Knowledge Prompt**

We trained this simple network on 100 epochs until it converged and we achieved an accuracy of 21% and a decreasing loss curve, as you can see in the figure.

You can see the accuracy and loss figure for this type of MLM task using injected knowledge in figure 4 and figure 7.

- **Training Results for a More Complex Neural Network as a Knowledge Prompt**

We trained this model on 80 epochs and the same hyperparameters as the previous model.

You can see the accuracy and losses for our more complex neural network in figure 5 and figure 8.

- **Training Results for an auto-encoder as a Knowledge Prompt**

We trained this model on 80 epochs using DropOut regularization and the activation function GELU.

You can see the accuracy and losses for our more complex neural network in figure 6 and figure 9.

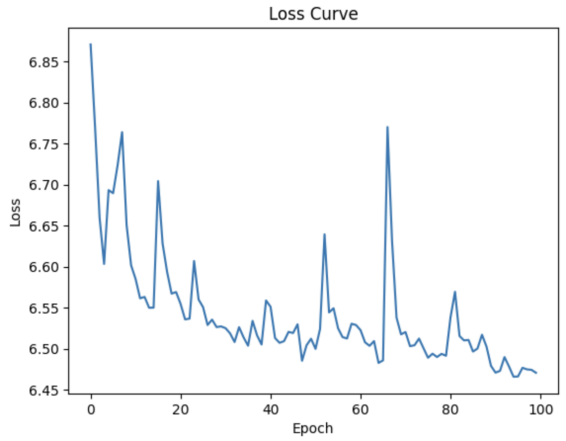

Figure 4: Zari-cda loss curve after training the basic Neural Network model as Knowledge Prompt

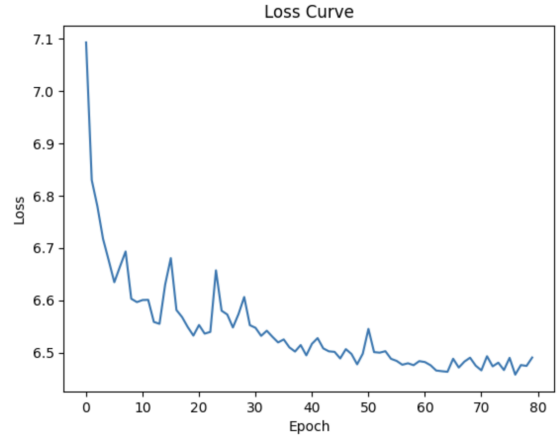

Figure 5: Zari-cda loss curve after training the more complex Neural Network model as Knowledge Prompt

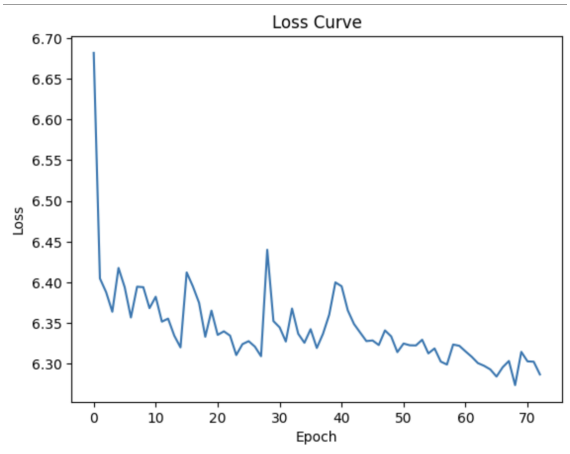

Figure 6: Zari-cda loss curve after training the AutoEncoder Network model as Knowledge Prompt

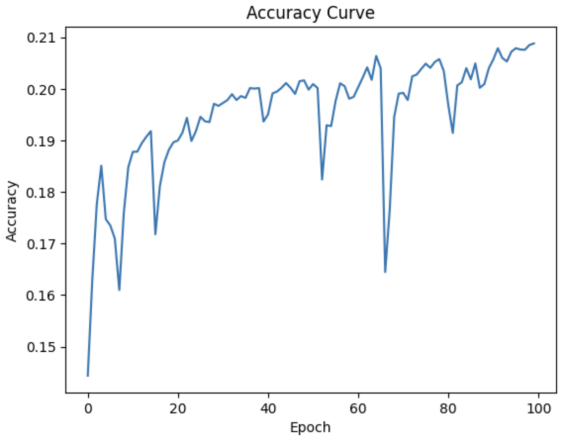

Figure 7: Zari-cda accuracy curve after training the basic Neural Network model as Knowledge Prompt

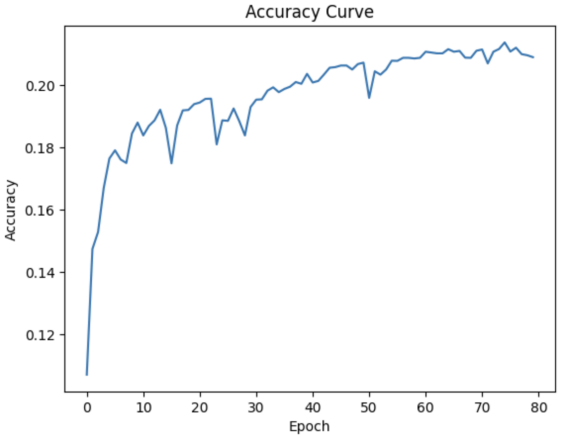

Figure 8: Zari-cda accuracy curve after training the more complex Neural Network model as Knowledge Prompt

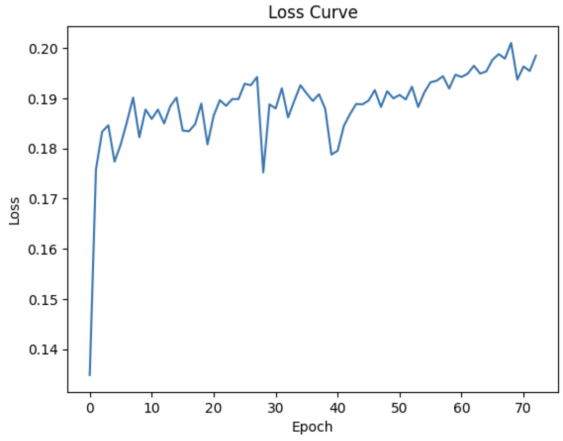

Figure 9: Zari-cda accuracy curve after training the AutoEncoder Network model as Knowledge Prompt

model	T1	T2	T3	T4	T5
zari-cda base	0.55	0.79	0.64	0.50	0.44
simple NN	0.33	0.51	0.37	0.29	0.22
comp NN	0.34	0.53	0.37	0.32	0.25
AE	0.36	0.51	0.40	0.31	0.23

Table 1: comparing the T1-T5 scores

model	Overal GS	GN	Bias
zari-cda base	0.59	0.75	0.66
simple NN	0.35	0.86	0.50
comp NN	0.36	0.86	0.51
AE	0.38	0.86	0.52

Table 2: comparing the overall bias scores.

4.2 Bias and Gender-Specific Knowledge Scores for Zari cda SP

The highest T1 score was achieved using the Auto Encoder prompt, with a score of 0.36. The highest T2 score was achieved using the complex Neural Net prompt, with a score of 0.53. The highest T3 score was also achieved using the Auto Encoder prompt, with a score of 0.40. The highest T4 score was achieved using the complex Neural Net prompt, with a score of 0.32. Finally, the highest T5 score was achieved using the complex Neural Net prompt, with a score of 0.25.

Overall, the complex Neural Net prompt performed best across all tests, achieving the highest scores on three out of five tests.

The highest overall gender-specific score was achieved using the complex Neural Net prompt, with a score of 0.86. The highest gender-neutral score was achieved using both the simple Neural Net and Auto Encoder prompts, with a score of 0.86. Finally, the highest bias score was achieved using the Auto Encoder prompt, with a score of 0.52.

Overall, the complex Neural Net prompt performed best in terms of overall gender-specific score, while both the simple Neural Net and Auto Encoder prompts performed best in terms of gender-neutral score. The Auto Encoder prompt had the highest bias score.

5 Discussion

In this project, we have explored the effectiveness of adding gender-specific information to debiased Zari model using knowledge prompts. We have found out that our methods were effective in re-introducing factual gender signals into the debiased model, while avoiding the re-learning of social biases in select models. We have achieved scores that were mentioned before in the report that were promising.

Our work results in a language model that is relatively not biased but has useful gender-specific information. This can enhance the performance and fairness of the language model on various natural language understanding and generation tasks, such as natural language inference, sentiment analysis, or text summarization.

We faced many challenges and had to do many trials and errors while figuring out how to use knowledge prompts to inject gender-specific knowledge. We tried different types of neural networks, objectives, and data sources for training the knowledge prompts. We also tried different ways of combining the knowledge prompts with the input of the language model. We learned from our failures and successes and improved our methods accordingly.

One of the important challenges that we had was the difference between the Stanza and BERT-large(Zari) tokenizers where they disagree on some tokens in the level of combining embeddings of entities that were recognized by Stanza and the input embeddings. For example, the Stanza tokenizer tokenized one word in 2 tokens while the BERT tokenizer tokenized that word as one token, and vice versa. This was a big challenge in the implementation but we successfully handled this problem.

The other challenge that we had was that Stanza can't be running parallel and that's what makes the training procedure long so we couldn't increase the training epochs. But there is an answer to this problem and that is available unofficial workarounds for running the Stanza paralell.

For future work, we could evaluate our model on downstream NLP tasks and compare its performance and fairness with other models that use different methods for debiasing or enhancing language models with gender-specific knowledge. We could also try different types of knowledge prompts, such as relation-centric or concept-centric ones, and compare their results with entity-centric ones. We could also explore other sources of world knowledge, such as Wikipedia or ConceptNet, and see how they affect the quality and diversity of the

knowledge prompts.

6 Conclusion

In this project, our objective was to come up with a methodology that would result in a relatively impartial model possessing cognizance of gender-specific information. To achieve this, we employed knowledge prompts to incorporate gender-specific data into the unbiased Zari model and assessed the efficacy of our approach using various metrics. In order to injecting gender-specific knowledge effectively, We examined the Zari-bert-cda model with 3 different levels of complexity of KP models. The results showed that by using more complex models as KPs, the model can better understand the complexity of the data structure and relations for gender-specific words, but in the convergence point of our 3 models, they couldn't reach the bias score of zari-bert-cda. We believe that more complexity of the model is needed to help to understand more gender-specific knowledge, as we can see in the results. Also, we believe that the data that we used is unbalanced and with the techniques like data augmentation, we can reach a better result of injecting the gender knowledge to the LM. You can see the distribution of data in Figure 10 where most of the labeled data is related to male gender-specific words.

The process of integrating gender-specific knowledge into the Zari model presented numerous challenges. We experimented with various neural networks, objectives, and data sources for training the knowledge prompts and explored different methods for combining the knowledge prompts with the language model's input. Through trial and error, we refined our approach.

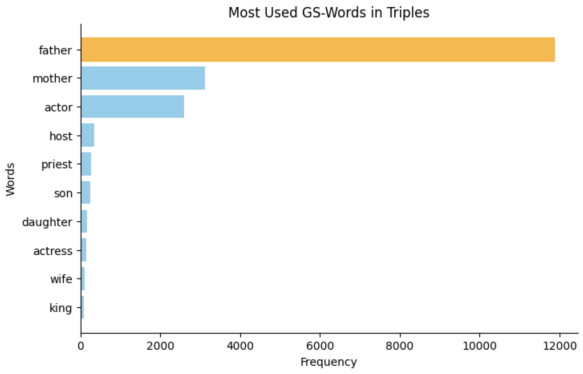

Figure 10: Most used gender-specific words in triples

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. *arXiv preprint arXiv:1607.06520*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Cicero Nogueira dos Santos, Zhe Dong, Daniel Cer, John Nham, Siamak Shakeri, Jianmo Ni, and Yun hsuan Sung. 2022. Knowledge prompts: Injecting world knowledge into language models through soft prompts. *arXiv preprint arXiv:2210.04726*.

Masahiro Kaneko, Danushka Bollegala, and Naoaki Okazaki. 2022. Debiasing isn't enough! – on the effectiveness of debiasing mlms and their social biases in downstream tasks. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1299–1310, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Yi Liu. 2021. A corpus-based lexical semantic study of Mandarin verbs of zhidao and liaojie. In *Proceedings of the 35th Pacific Asia Conference on Language, Information and Computation*, pages 646–651, Shanghai, China. Association for Computational Lingustics.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: A free collaborative knowledge base. *Communications of the ACM*, 57(10):78–85.

Mahdi Zakizadeh, Kaveh Eskandari Miandoab, and Mohammad Taher Pilehvar. 2023. "difair: A benchmark for disentangled assessment of gender knowledge and bias. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*.

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2018. Gender bias in coreference resolution: Evaluation and debiasing methods. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 15–20, New Orleans, Louisiana. Association for Computational Linguistics.

A Appendix

The list of our gender-specific words: 'actors', 'actresses', 'airmen', 'boys', 'girls', 'brothers',

'sisters', 'businessman', 'businessmen', 'chairman', 'chick', 'dude', 'chicks', 'daddy', 'mommy', 'daughters', 'sons', 'fathers', 'mothers', 'sons', 'daughters', 'gal', 'guy', 'guys', 'gentleman', 'lady', 'gentlemen', 'ladies', 'males', 'females', 'kings', 'queens', 'king', 'queen', 'actor', 'actress', 'waiter', 'waitress', 'son', 'daughter', 'uncle', 'aunt', 'nephew', 'niece', 'prince', 'princess', 'brother', 'sister', 'father', 'mother', 'spokesman', 'wives', 'husbands', 'men', 'women', 'duke', 'dukes', 'emperor', 'empress', 'emperors', 'landlord', 'master', 'mistress', 'masters', 'monk', 'nun', 'monks', 'nuns', 'priest', 'priestess', 'priests', 'sorcerer', 'steward', 'wizards', 'witches', 'fraternity', 'sorority', 'groom', 'bride', 'heroine', 'hero', 'host', 'hostess', 'husband', 'wife', 'lad', 'las', 'lord', 'lady', 'masculine', 'feminine', 'paternal', 'maternal', 'widow', 'wizard', 'witch', 'grandfather', 'grandmother', 'boyfriend', 'girlfriend', 'fiancé', 'fiancée', 'penis', 'prostate', 'seminal', 'sperm', 'pregnant', 'pregnancy', 'cervical'

The words the were present in the wikidata triples: 'businessman', 'brother', 'emperor', 'king', 'monk', 'hero', 'kings', 'husband', 'actor', 'chairman', 'host', 'uncle', 'sons', 'men', 'brothers', 'prince', 'father', 'wizard', 'dukes', 'son', 'lord', 'duke', 'master', 'priest', 'grandfather', 'nephew'] female=['wife', 'mother', 'las', 'grandmother', 'mistress', 'witch', 'wives', 'heroine', 'sister', 'sorcerer', 'women', 'nun', 'actress', 'princess', 'widow', 'queen', 'daughters', 'empress', 'witches', 'maternal', 'sisters', 'daughter'.