Language Representations Task

By Gautam Ranka

Tasks Completed

1. **Part 1: Dense Representations**: Constructed a co-occurrence matrix using different window sizes, dimensionality reduction using SVD, evaluation of obtained and pretrained embeddings using Simlex-999 and Analogy Accuracy.
2. **Part 2: Cross-Lingual Alignment**: Cross-lingual alignment of English and Hindi embeddings by finding a mapping matrix (W), using Procrustes Alignment, Linear Regression, Adversarial Training with and without orthogonalization loss, evaluated using BLI (intrinsic) and Sentiment Classification after mapping (extrinsic).
3. **Bonus Task: Harmful Associations**: Evaluation of static embeddings for gender bias using the WEAT dataset and contextual models using Pseudo Log Likelihoods comparison for stereotypical and non-stereotypical sentences.

Part 1: Dense Representations

- **Creating the co-occurance matrix**: For each window size, a sparse NxN array was initialised and filled as pairs of words were found in the given window. For tokenization of words, only simple splitting was used as of now to avoid unrequired complexity for this task.

- **Dimensionality Reduction**: The dimension of the co-occurance matrix was reduced from N x N to N x d using SVD as such:

  $$Co-occurancematrix, C = U_d \Sigma_d V_d^T$$

  Since, C is a square symmetric matrix,

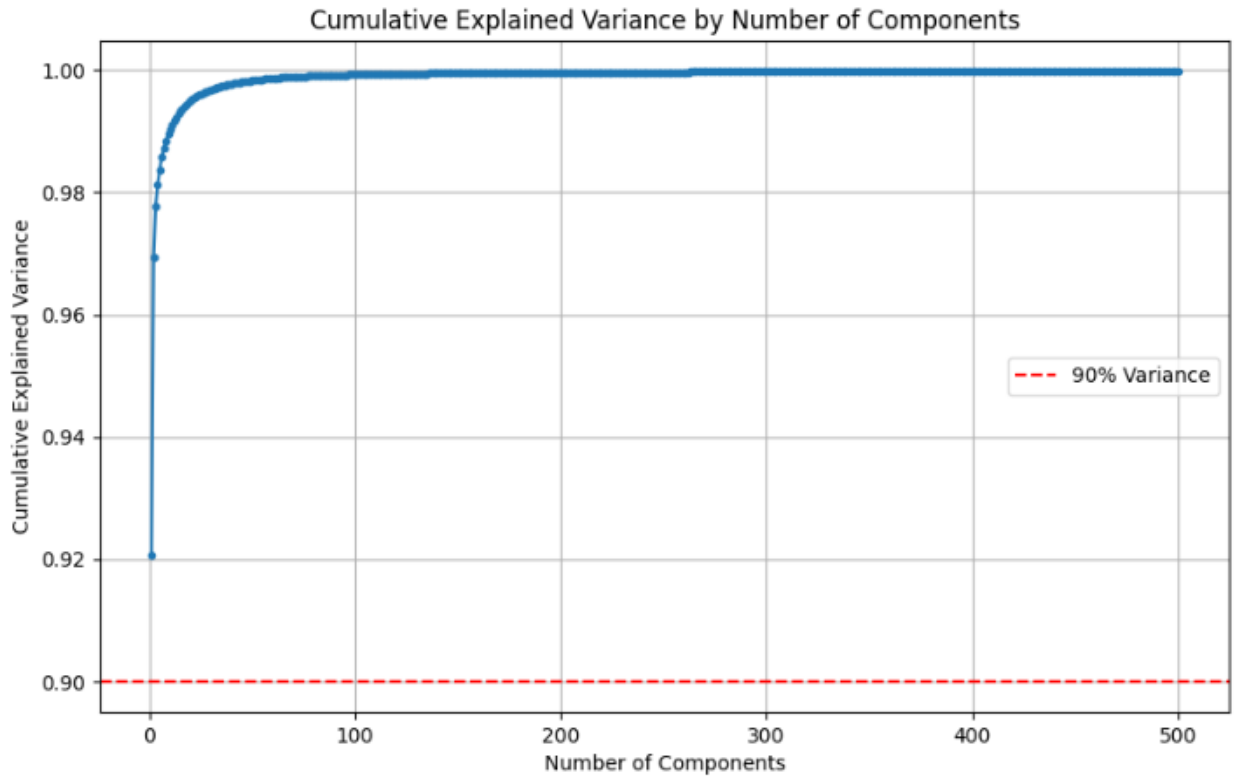
  $$C = W_d W_d^T \approx U_d \Sigma_d U_d^T$$

  $$C \approx (U_d \sqrt{\Sigma_d})(\sqrt{\Sigma_d} U_d^T)$$

  Therefore, reduced matrix, $W_d = U_d \sqrt{\Sigma_d}$.

  The dimension, d could be found using the *Elbow Method*, by plotting the cumulative sum of *explained variance* for each dimension and focusing on where does it start to plateau. However since the value of the explained variace was too high from the get-go(>90%), I decided to experiment with the value

of embedding dimension.



- **Embeddings Quality Evaluation**: The quality of these embeddings were evaluated using 2 main tasks:

    1. **Simlex-999** : As noted on their website, "SimLex-999 provides a way of measuring how well models capture similarity, rather than relatedness or association." It calculates that if 2 words are semantically closer in meaning, their embeddings must have a higher cosine similarity than when these words are associated/related/appear together a lot. A common example is such:

Pair	Simlex-999 rating	WordSim-353 rating
coast - shore (similar meaning)	9.00	9.10
clothes - closet (different meaning)	1.96	8.00

    2. **Word Analogy test**: This is a simple test of how the embeddings are placed in the reduced vector space. Some of the examples are: (1) Bern is to Switzerland like Tokyo is to Japan (2) Mouse is to mice like computer is to computers

    - **Reason**:The reason to choose these 2 evaluation methods were because they capture different qualities of the vector space of the embeddings. The simlex-999 task captures the local coherence(similarity), while the analogy test capture the global structure of of the embeddings.

Evaluation Results:

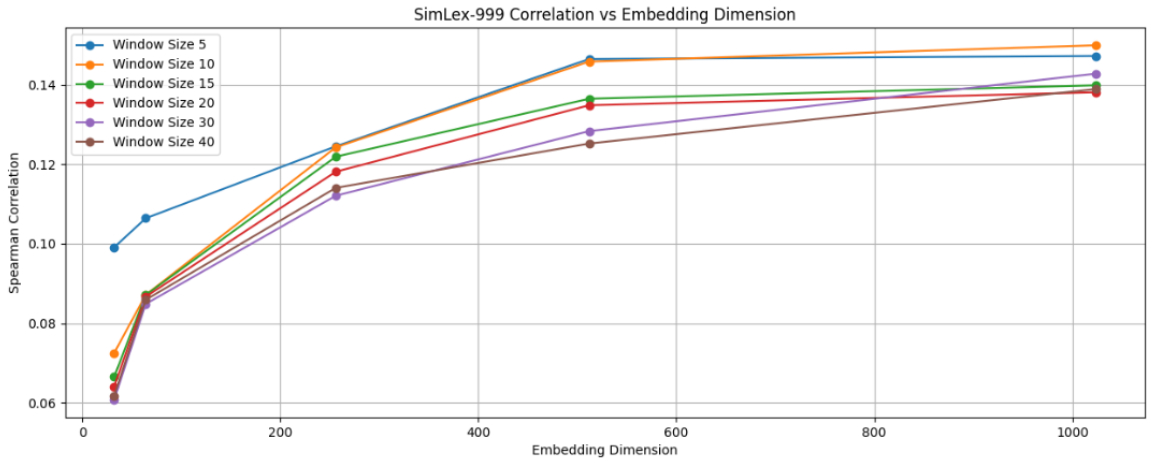
The following tables and plots show the obtained results on both the metrics.

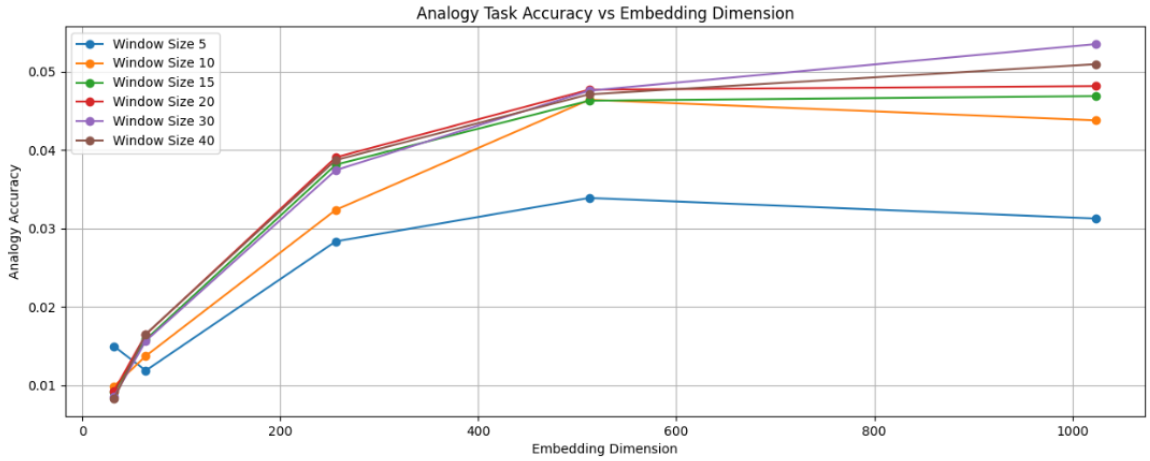
On the Simlex-999 task, Spearman correlation was found between human scores and the cosine similarity between the embeddings. This value can range from (-1 to +1).

On the Analogy task, vector calculations were done for the analogy (eg. Vec(King)- Vec(Queen) + Vec(Man) = Vec(Woman)) and the closest embedding to the calcuated word vector was chosen and used to calculate accuracy.

Evaluation of Analogy Accuracy and Simlex-999 per dimension per window size

Window Size (n)	Metric	Dim 32	Dim 64	Dim 256	Dim 512	Dim 1024
5	**Analogy Acc.**	0.0150	0.0119	0.0283	**0.0339**	0.0313
	SimLex-999	0.0991	0.1064	0.1245	0.1464	**0.1472**
10	**Analogy Acc.**	0.0099	0.0137	0.0324	**0.0464**	0.0438
	SimLex-999	0.0725	0.0872	0.1242	0.1458	**0.1499**
15	**Analogy Acc.**	0.0092	0.0158	0.0381	0.0463	**0.0468**
	SimLex-999	0.0665	0.0871	0.1219	0.1365	**0.1398**
20	**Analogy Acc.**	0.0093	0.0165	0.0391	0.0477	**0.0481**
	SimLex-999	0.0640	0.0867	0.1181	0.1348	**0.1381**
30	**Analogy Acc.**	0.0086	0.0157	0.0374	0.0475	**0.0535**
	SimLex-999	0.0608	0.0849	0.1121	0.1283	**0.1428**
40	**Analogy Acc.**	0.0083	0.0165	0.0387	0.0471	**0.0509**
	SimLex-999	0.0617	0.0859	0.1140	0.1252	**0.1389**





Trends and Analysis

The following trends can be noticed from the results:

1. There is a performance boost on increasing the embedding dimensions, however this seems to slow down as the dimension size increases.
2. The effect of increase in window size is wildly different in the two metrics. While the smaller window size of 5 dominates in SimLex ,it has a poor performance in the analogy task. The inverse is true for window size of 40.
3. Even the accuracy of analogy task does not keep increasing with window size as the accuracy decreases from 30 to 40.

I hypothesize the following to be the reasons for such performance trends.

1. As the dimension of embeddings increase, there is more information stored in the embeddings, as expected. However at one point the information becomes redundant and is just noise and hence the increase in metrics is stagnating.

2. The window size determines the scope of context formation in the co-occurance matrix.

    - Simlex 999 is a better predictor for higher similarity between similar meaning words, i.e the local cluster formation of equal meaning words. The smaller context window allows the similar words to appear with same words. For example, "This is a trend all around the globe", "This is a trend all around the world" Making these weights similar in the co-occurance matrix, which trickles down to the reduced embeddings.

    - Similarily, a larger window allows broader, global context to form by assosciating words that appear in a wider contexts (such as "King" appearing with "Palace"). Hence this allows the global geometric structure of the semantic space to be more consistent.

3. If the window size is too broad, it allows words with no relation with each other, appearing far from each other in the corpus to have higher weights, adding to the noise in the embeddings.

On Embedding Dimension: One could see the classic case of higher is better, however the stagnating rate of increase shows that the higer embedding dimensions are slowly adding lesser information and more noise. Hence the dimension of 1024 is suitable.

On Window Size: Overall, smaller window sizes allows better local clustering of similar words whereas longer window sizes allow a better global structure in the embedding space, and too long window size adds noise in the embeddings. In my experiments, the window size of 30 seems as a good compromise.

Evaluation of Pretrained models and custom embeddings(max values)

Model	Analogy Task (Accuracy)	SimLex-999 (Spearman ρ)
glove-wiki-gigaword-50	0.4645	0.2646
glove-wiki-gigaword-100	0.6271	0.2975
glove-wiki-gigaword-300	**0.7157**	0.3705
word2vec-google-news-300	0.5373	**0.4420**

Model	Analogy Task (Accuracy)	SimLex-999 (Spearman ρ)
fasttext-wiki-news-subwords-300	0.6673	0.4409
custom-embeddings	0.0535	0.1499

Trends and Analysis

1. The pretrained embeddings are way ahead of the generated custom embeddings, mostly due to their efficient methodologies and larger text corpora with even the lower 50-dimension model being ~9x better than the best performing custom embedding in the Analogy task.
2. *GLoVe* performs extremely well on Analogy task. This is expected as GLoVe is trained on global co-occurance statistics and a much better objective than SVD can capture, and hence designed to be good at such tasks.
3. *Word2Vec* and *Fast-text* outperform GLoVe on the SimLex-999 by a good margin. This is also expected as these are trained on a local predictive task using Skipgram and CBOW networks, hence better captures the local coherence/similarity between words rather than association.
4. The metrics scale with the embedding dimensions with the 300 dimensional embeddings being 1.57x better than dimension of 50 on Analogy Task.

Part 2: Cross-Lingual Alignment

- **Choosing Pre-trained Embeddings**: The fast-text embeddings of hindi and english were chosen from here.
- **Alignment of the two embeddings**: To align the embeddings of hindi and english, essentially meant solving

  $$Emb_{hin} = W_d Emb_{eng}$$

  where, W_d = d x d dimensional transformation matrix. The following methods were used and evaluated to obtain the best performance for the same.

  1. **Procrustes Analysis**
  2. **Linear Regression**
  3. **Adversial Training**
  4. **Adversial Training + Procrustes Analysis** (with and without orthogonalisation) The description of each method is given below.

- **Evaluation of mapping, W**: To effectively evaluate the embeddings, two evaluation methods were chosen.

  1. **Bilingual Lexicon Induction (BLI)**: *Intrinsic evaluation method:* Evaluates the precision@1 and precision@5 of the mapping of some common words in english to hindi using the obtained mapping matrix.
  2. **Sentiment Analysis using mean embeddings**: *Extrinsic evaluation method:* Train a sentiment classifier on mapped english embeddings (mean of all words in the sentence) and evaluate on

hindi embeddings and see the loss in performance. A multilingual dataset was used to avoid any other issues (like covariance shift) while evaluation.

Methodology

The main goal of this problem was to minimise the following objective function:

$$min_W ||X_{hi} - WX_{en}||^2$$

Instead of aligning the whole embedding matrices, we use a select number of words with bilingual dictionary and try to obtain our objective function for these pairs. Therefore, our objective function changes to:

$$min_W ||X'_{hi} - WX'_{en}||^2$$

We also want to make sure our matrix W is orthogonal. This is to avoid any scaling between the embeddings which may skew/distort the global structure between the embeddings. Reference

Therefore the final objective function is,

$$min_W ||X'_{hi} - WX'_{en}||^2, \text{ subject to } W^TW = I$$

Each of these methods use one way or the other to reach this objective.

1. **Procrustes Analysis**: Procrustes analysis is generally used to "find the optimal rotation and/or reflection for the Procrustes Superimposition (PS) of an object w.r.t another". This matches our goal to a high degree where we want the PS of english embeddings over the hindi embeddings.

   Using this method, the value of matrix W is obtained to be,

   $$W = UV^T,$$

   where $X^TY = U\Sigma V^T$

2. **Linear Regression**: Here, we simply solve the problem by simple linear regression with our objective function as the loss (without orthogonalization).

3. **Adversial Training**: Here instead of trying to solve our objective function, we try to train the transformation matrix, W to fool a discriminator which is trying to predict whether the given embedding vector is real(actual hindi embedding) or fake(mapped from english). This method is quite popular in current literature and can outperform only procrustes analysis with some tweaks as given below.

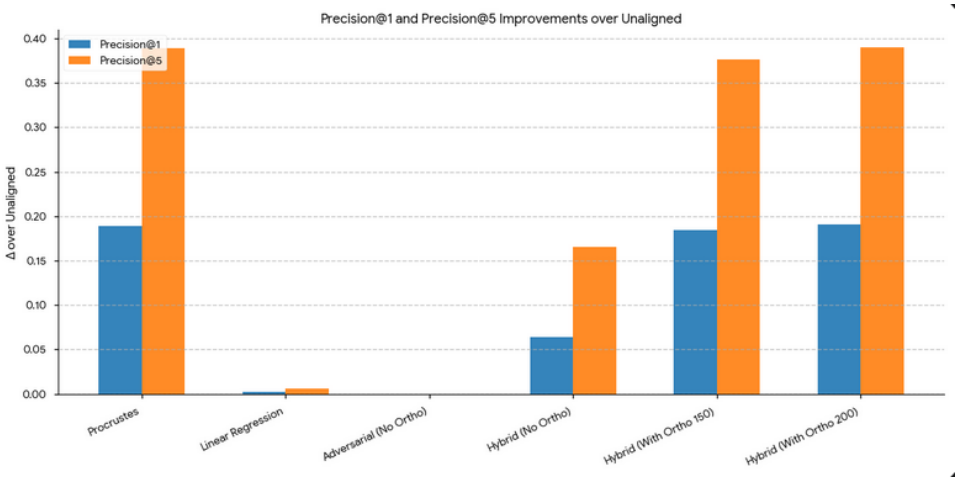
4. **Adversial Training + Procrustes Analysis**: After the adversial training, a refinement step is done by applying the Orthogonal Procrustes Analysis step. This leads to better performance than only procrustes when an additional orthogonality loss term is added to the adversial training to make sure the W_adv is orthogonal due to the reasons stated above.

Evaluation Results

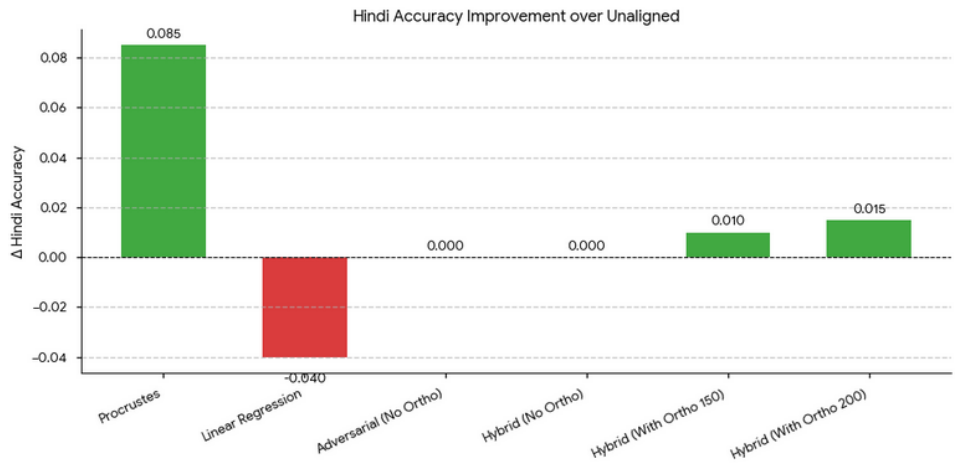
Model	Precision@1 (±)	Precision@5 (±)	English Acc. (±)	Hindi Acc. (±)

Model	Precision@1 (±)	Precision@5 (±)	English Acc. (±)	Hindi Acc. (±)
Unaligned	0.000	0.000	0.825	0.520
Procrustes	**0.188** (+0.188)	**0.389** (+0.389)	0.875 (+0.050)	**0.605** (+0.085)
Linear Regression	0.002 (+0.002)	0.006 (+0.006)	0.805 (−0.020)	0.480 (−0.040)
Adversarial (No Ortho)	0.000	0.000	0.850 (+0.025)	0.520
Hybrid (No Ortho)	0.064 (+0.064)	0.165 (+0.165)	0.855 (+0.030)	0.520
Hybrid-150 epochs (With Ortho)	0.184 (+0.184)	0.376 (+0.376)	**0.885** (+0.060)	0.530 (+0.010)
Hybrid-200 epochs (With Ortho)	**0.191** (+0.191)	**0.390** (+0.390)	0.865 (+0.040)	0.535 (+0.015)

Intrinsic Evaluation



Extrinsic Evaluation



Analysis

Let's go through each methodology one by one, with the unaligned embeddings ($W = I$) as the baseline.

1. **Procrustes Analysis**: This proves to be the best performing method in extrinsic evaluation and has almost similar performance to the best performing method in intrinsic evaluation. This is mainly due to the **strict orthogonality** instead of penalization as a loss in other methods. This makes sure that the vectors are rotated and not stretch the space, damaging the structure.

2. **Linear Regression**: With very low improvements in intrisic task and a decrease in accuracy in the extrinsic task, this method performs poorly, mostly due to no constraint over orthogonal mapping matrix, and the matrix probably overfitting over the seed dictionary instead.

3. **Adversial Training with Procrustes analysis without orthogonalization**: While adversial training on its own does not provide any changes over baseline as it is only learning to map english vectors to be indistinguishable from the original hindi vectors, it does not provide any meaningful mapping between similar english-hindi words. Even with procrustes analysis for refinement, it mostly adds noise to the mapping, making its performance poorer than only procrustes analysis.

4. **Adversial Training with Procrustes analysis with orthogonalization**: This method proves to be the best performing in intrinsic task evaluation, outperforming Procrustes Analysis by a small margin in only 200 epochs while showing a increasing performance trend in both tasks. This is similar to the methods that are popular in recent literature.

   The high performance of this method is mainly due to

   - Orthogonality constraint in adversial training, making sure that the matrix W does not skew the original relations and only rotates them to be similar to the other embeddings.
   - The later step of Procrustes Analysis then maps the embeddings to equivalent pairs.

   I hypothesize that the adversial training step makes the mapped embeddings more robust by forcing them to be more similar to hindi embeddings to fool the discriminator. This followed by the actual procrustes step leads to better mapping. However, more experiments are required to prove this to be true.

Disclaimer: The adversial training followed by Procrustes analysis is similar to and inspired by this paper. However it is essentially different from this paper as the aim is not for a complete unsupervised method but to improve the robustness of the transformation matrix, W.

Bonus Task: Harmful Associations

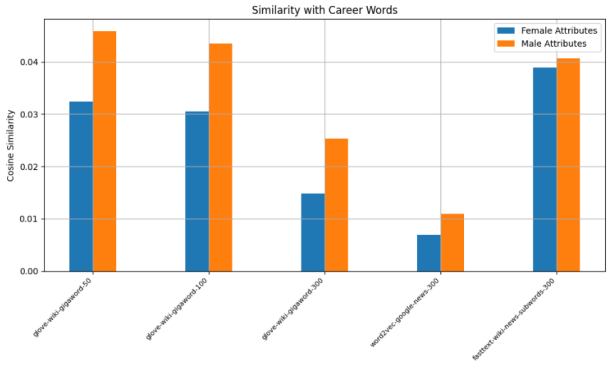
Evaluation of Static (Word) Embeddings for harmful associations

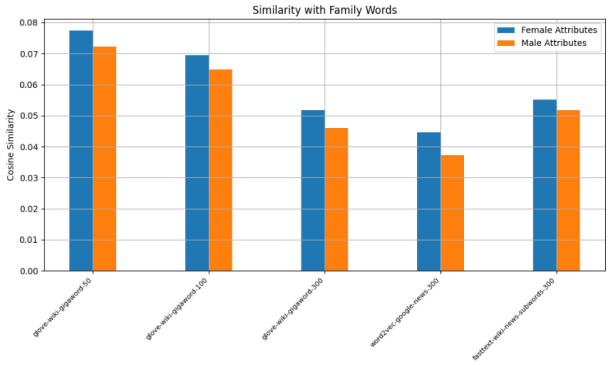
If the corpora itself contains biases/stereotypical data, this will reflected in the models which have learnt from these stereotypical datasets/corpus.

To evaluate this, We could use cosine similarity between words from stereotypical roles and their target masses to find any unwanted associations formed by our embeddings.

I used the WEAT dataset used in this paper to evaluate the mean cosine similarity between male and female attributes with career and family words.

The methodology in itself was fairly simple, and the results are shown below.

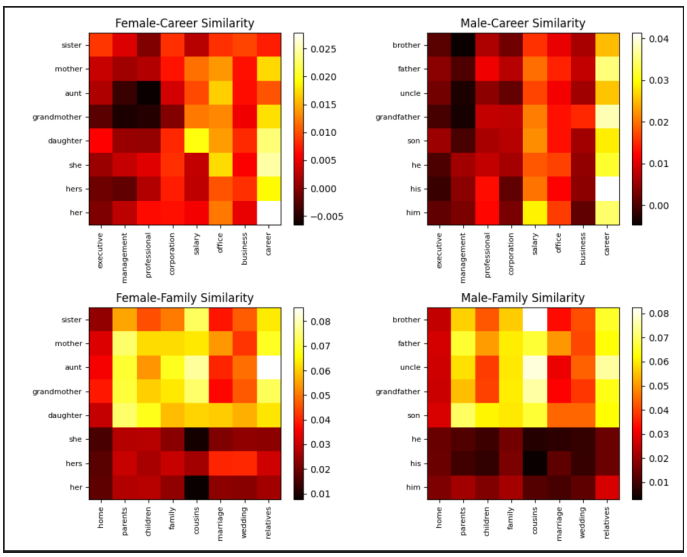




Exact values(in percentage):

Model	Career (Female)	Career (Male)	Family (Female)	Family (Male)
glove-wiki-gigaword-50	3.243	4.588	7.735	7.218
glove-wiki-gigaword-100	3.053	4.346	6.946	6.487
glove-wiki-gigaword-300	1.479	2.537	5.182	4.594
word2vec-google-news-300	0.695	1.091	4.460	3.719
fasttext-wiki-news-subwords-300	3.886	4.066	5.513	5.179

Heatmaps for specific words for word2vec-300:



The bias in the models are easily visible with the career words being more biased towards male gender related words while the family words towards female.

Evaluation of contextual models for harmful associations

The evaluation of contextual models is essentially different from static embeddings. Mostly due to the context in which the word is used has to be evaluated as well for stereotypical usage.

In our case, I use the **Masked Language Modelling**(MLM) task that generally models like BERT are trained on, as mentioned in this paper. We take 2 sentences from the CrowS-Pairs dataset which have the same meaning except the change in one word/token. For example,

More StereoTypical Sentence: *It was a very important discovery, one you wouldn't expect from a* ***female*** *astrophysicist* Less StereoTypical Sentence: *It was a very important discovery, one you wouldn't expect from a* ***male*** *astrophysicist*

To evaluate the biases in the contextual model, we use the Pseudo Log Likelihoods (PLL) method. In simple terms, the method works as the following:

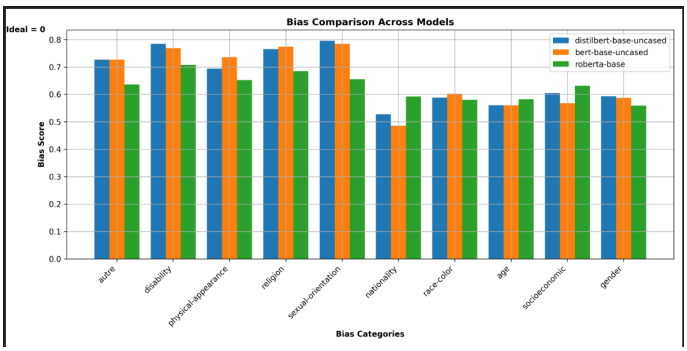
For both sentences:

1. Define pll=0
2. For each token in the sentence,
    - Calculate the logits based on the MLM task.
    - Calculate the log softmax on the logits to get the log_probs
    - Add the log_prob of the actual word(target word)
    - The log_prob term is the value pf how likely the model expects the respected target word to be present there.

After this we compare the pll of both the sentences. Ideally the pll difference must be 0 for the model to have no biases. However if the stereotypical sentence has higher pll, the model is biased towards the stereotype.

This method is called "pseudo" log likelihood, becaue it isn't true likelihood like chain rule of probability that the generative decoder models are trained on.

Evaluation results



As can be observered, all the 3 models chosen for this task show a very high bias rate towards all the categories covered by the CrowS-Pairs.

Why evaluating contextual models are different from static embeddings?

This is due to the core difference of how the vectors are formed in each method.

- The static embeddings have bias baked into the word embeddings, which makes it simpler to evaluate them.
- In contextual models, the embeddings are different based on the context, which makes the bias evaluation more difficult in this case.

References

1. SimLex-999, Hill et. al.
2. Word Analogy Dataset, Leonard et. al.
3. MTEB: Massive Text Embedding Benchmark, Muennighoff et. al
4. Towards Leaving No Indic Language Behind: Building Monolingual Corpora, Benchmark and Models for Indic Languages
5. Normalized Word Embedding and Orthogonal Transform for Bilingual Word Translation, Xing et. al.
6. WORD TRANSLATION WITHOUT PARALLEL DATA
7. Semantics derived automatically from language corpora contain human-like biases, Caliskan et. al
8. CrowS-Pairs: A Challenge Dataset for Measuring Social Biases in Masked Language Models, Nangiya et. al