Machine Learning for NLP - Neural Network Report

Goya van Boven 2743245

January 2022

1 Introduction

In this report I investigate the task of automated sexism recognition using Neural Networks (NN). To train the models I use the Sexist Workplace Statements Dataset (Grosz & Conde-Cespedes, 2020), which contains sentences expressed at the workplace, annotated for the presence of sexism. Glick & Fiske (1996) distinguish between a *benevolent* and *hostile* sexism, where the former is more positive and benign in appearance, while both forms carry paternalism, as well as stereotypes and assumptions about women, their role in society and their sexuality. Jha & Mamidi (2017) further extend this definition by claiming benevolent sexism is more subtle whereas hostile sexism is more explicitly negative. Grosz & Conde-Cespedes (2020) argue that according to this distinction sexism at the workplace can often be categorized as benevolent. Continuing, the authors claim sexism can be considered to be a form of hate speech. In the field of automated hate speech detection however, many datasets are based on text from social media (Waseem & Hovy, 2016; Samory et al., 2021; De Gibert et al., 2018; Qian et al., 2019): a context in which sexism tends to be of a more hostile nature. But since sexism too often remains a part of work environments, (Krivkovich et al., 2017) it is important that this kind of benevolent sexism can be correctly recognized as well.

In this study I train two baseline models, a logistic regression model and a Support Vector Machine (SVM), to which I compare the performance of the Neural Networks. To build the NNs I adapt the code from Rashid (2016). I experiment with various word embedding reperesentations, namely TF-IDF, Word2Vec (Mikolov, Sutskever, et al., 2013; Mikolov, Chen, et al., 2013), and GloVE (Pennington et al., 2014). Additionally, I perform hyperparameter tuning in which I optimize the learning rate, number of epochs and the number of hidden nodes. I find the best performing Neural Network with GloVe embeddings, a learning rate $\alpha = 1e^{-3}$, 768 hidden nodes and 40 epochs. This model achieves an accuracy of 82.9 on the test set, an improvement of 6.6 over the logistic regression model (accuracy = 76.3), and of 7.5 over the SVM (accuracy = 75.4).

This report is structured as follows. The data and its preprocessing are described in Section 2. Next, I describe the methodology (Section 3) and present the results (Section 4). Finally I analyze common errors in Section 5.

2 Data

In this study I use the Sexist Workplace Statements Dataset. This dataset contains 1100 sentences that can be expressed in the context of the workplace, which are annotated for containing sexism (1) or being ambiguous or neutral (0). These sentences are taken from tweets, quotes from the press and quotes submitted by students and faculty members. The sentences were annotated by the authors (Grosz & Conde-Cespedes, 2020), and no strict definition of sexism was followed. Examples of sentences labelled as sexist are the following:

- *We know why she was hired...*
- *Are you having period ?*
- *A secretary must think like a man act like a lady look like a girl and work like a dog.*

Examples of neutral or amiguous sentences are

- *You are a human not a robot, take care !*
- *My soul, sir? I haven't got one. The management doesn't allow them.*
- *Should a guy open the car door for a girl?*

In the dataset, 624 sentences are labelled to be sexist (55.9%) and 513 are labelled to be ambiguous or neutral (45.1%). The ratio of sexist/neutral sentences is thus rather balanced in the dataset. This ratio does not realistically reflect the real-word situation, where in most cases a considerately smaller rate of sentences will show sexism.

In order to preprocess the data I shuffled it randomly and split it into a 60% train set, 20% dev set and 20% test set. I then tokenized the sentences and lemmatized the tokens using NLTK (Bird et al., 2009).

3 Methods

3.1 Sentence vectors

Sexism emerges on a sentence level, rather than on a word level. Therefore, the model input vectors should be sentence-, rather than word representation. I compare several methods to obtain such vectors:

- Count-based TF-IDF, which takes the co-occurences of terms in the corpus and weights them based on (i) the log frequency of the term in the document (Luhn, 1957) multiplied by (ii) the log of the inverse document frequency, i.e. the number of documents in the corpus that the term occurs in. In this study each individual sentence is considered as being a document when computing the TF-IDF weight. Continuing I use Singular Value Decomposition, a method that finds the most important dimensions in a matrix, to reduce the dimensionality of the vectors to 300.

- Pre-trained dense Word2Vec (W2V) embeddings (Mikolov, Sutskever, et al., 2013; Mikolov, Chen, et al., 2013). More specifically, I use the embeddings trained on the 100 billion tokens Google News corpus (Mikolov, Sutskever, et al., 2013) using the Continuous Bag of Words implementation. In this implementation context words are used as input to a Neural Network that predicts a target word: the word that appears in the middle of the context in the training data. After the training process is completed the learned model weights are used as word vectors. The embeddings have a dimensionality of 300.
  I compare two ways of combining the individual word vectors into representations for the entire sentence: (i) summing over the word vectors and (ii) taking the mean of the word vectors.
- Pre-trained GloVe embeddings (Pennington et al., 2014). This method captures both global corpus statistics, i.e. word co-occurrence scores, and local statistics, i.e. local word context information. Specifically, I use 300 dimensional embeddings trained on a 840 billion tokens Common Crawl corpus (Pennington et al., 2014). For these embeddings I again compare summing over and taking the mean of the individual vectors to create sentence representations.

In total I thus compare five methods: TF-IDF, W2V$_{Mean}$, W2V$_{Sum}$, GloVe$_{Mean}$, GloVe$_{Sum}$. I expect the W2V and GloVe methods to outperform TF-IDF, as the former two methods have been shown to capture complex semantic relations (Mikolov, Sutskever, et al., 2013; Mikolov, Chen, et al., 2013; Pennington et al., 2014) and are trained on much larger corpora than the Sexist Workplace Statements Dataset, from which the TF-IDF vectors are obtained. As Grosz & Conde-Cespedes (2020) use GloVe embeddings in their models, I expect these to also produce the best results in the current experiments.

3.2 Models

I create three different types of binary classification models, in which the task is to predict whether an input sentence contains sexism (1) or not (0). In all models, the sentence vectors are the only input features used. Firstly, I create two baseline models: a Support Vector Machine (SVM) with a linear kernel and a logistic regression model. I train both baseline models on the TF-IDF vectors. Next, I create a Neural Network using the code of Rashid (2016) as a basis. This NN has 300 input nodes, one hidden layer (of which I vary the number of nodes in Experiment 2), and one output node. I make two adaptations to the Neural Network code. Firstly, I include precision, recall and F1-score as performance measures. Second, I add a `verbose` option to the training process, that prints after each epoch (i) how much time has elapsed, (ii) the performance on the training set and (iii) the performance on the dev set.

3.3 Experimental setup

I carry our three experiments in total. In the first two experiments, I randomly initialize and train the Neural Network five times, and report the mean performance scores of these trials.

Experiment 1 I train the Neural Network using the five different methods described above (Section 3.1) to obtain sentence vectors. I use the following hyperparameter settings: learning rate = 0.0005, number of hidden nodes = 128, number of epochs = 20. For this experiment I report the model performances on the dev set.

Experiment 2 Using the sentence vectors that worked best in the previous experiment as input, I now continue to tune the following hyperparameters:

1. The number of hidden layers k, with $k \in \{32, 64, 128, 256, 512, 768, 1012\}$;
2. The learning rate α, with $\alpha \in \{1e^{-5}, 5e^{-5}, 1e^{-4}, 5^{-4}, 1e^{-3}, 5e^{-3}, 1e^{-2}\}$;
3. The number of epochs n, with $n \in \{1, 5, 10, 15, 20, 30, 40\}$.

I perform the tuning in that order, initially using the same hyperparameters as in Experiment 1, but changing after each round to the best value of that hyperparameter. In this experiment, I again compare and report on the model performances on the dev set.

Experiment 3 In the final experiment I train the two baseline models and compare the performance of these two models to the performance of the best NN. In this experiment I compare of the Test set.

In the experiments I consider the evaluation metrics accuracy, precision, recall and F1-score. As the dataset is rather balanced, accuracy gives a good reflection of the performance of the model. Of the other metrics, I consider recall to be the most important, as missing a sexist expression can be more damaging than incorrectly identifying something as sexist.

To implement the SVM, the logistic regression model and the TF-IDF vectors I use Scikit learn (Pedregosa et al., 2011).

4 Results

Experiment 1 Table 1 shows the results of Experiment 1, in which various methods for obtaining sentence vectors are compared. The GloVe$_{Mean}$ model obtains the best result in terms of accuracy, precision and F1-score, while the TF-IDF model obtains the best recall. Note however that although the TF-IDF model obtains a close to perfect recall score, it's precision and accuracy are around 55, which corresponds to the percentage of sexist sentences (i.e. the sentences annotated with 1) in the dataset. The model could thus obtain this score by predicting 1 for nearly all sentences.

The GloVe$_{Sum}$ model achieves the same F1-score as the GloVe$_{Mean}$ model even though its precision is lower, because it achieves a higher recall than GloVe$_{Mean}$. Although I consider recall to be more important than precision for this task, I still prefer GloVe$_{Mean}$ as it reaches a higher accuracy than GloVe$_{Sum}$.

The fact that the GLoVe models outperform the other models is according to expectations. Also according to expectations, the TF-IDF model performs the worst (in all measures except recall). While for GloVe taking the mean over the word embeddings appears to work better, the reverse is true for W2V, where the *Sum* version outperforms its *Mean*-counterpart.

Method	**Accuracy**	**Precision**	**Recall**	**F1-Score**
TF-IDF	58.3	55.0	**98.4**	70.5
W2V$_{Mean}$	69.0	64.3	89.2	74.5
W2V$_{Sum}$	80.8	74.5	94.4	83.3
GloVe$_{Mean}$	**83.2**	**79.3**	90.4	**84.5**
GloVe$_{Sum}$	82.1	75.5	95.9	**84.5**

Table 1: Performance scores on the dev set for the 5 sentence vector methods (Section 3.1) compared in Experiment 1. Scores are the mean of 5 random initializations.

Experiment 2 As the GloVe$_{Mean}$ model achieved the best result in the previous experiment, I continue to use these embeddings for training the the upcoming models. Figure 1 shows the accuracy for the number of hidden layers I experimented with. Here one can see that the best performance - an accuracy of 83.7 - is achieved by the model that includes 768 hidden nodes. This model also achieves good results for the other metrics (precision = 78.8, recall = 92.9, F1-score = 85.2).

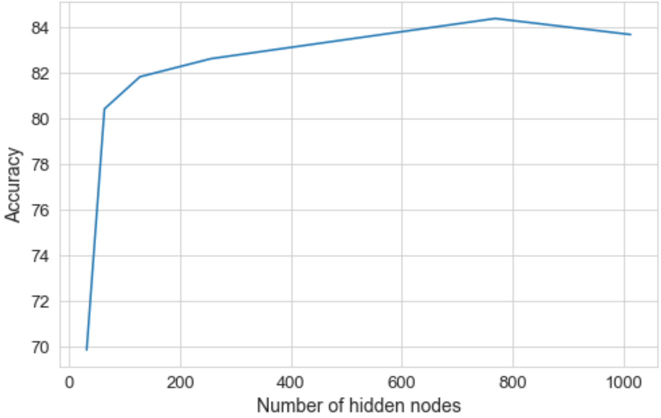

Figure 1: Model accuracy on the dev set per number of hidden layers

Next, I tune the learning rate α. Here, I continue to use 768 hidden nodes. The results can be found in Figure 2. This figure shows that increasing the learning rate benefits the accuracy up to $\alpha = 1e^{-3}$, which gives accuracy = 85.5, while increasing it further harms the accuracy score. This learning rate also achieves the highest recall (94.4) and F1-score (86.8), and achieves a good precision (80.3). The best precision (of 85.1) is achieved by a learning rate of $1e^{-2}$, but this model performs worse on all other metrics (accuracy = 80.5, recall = 74.6, F1-score = 79.5). I therefore consider $\alpha = 1e^{-3}$ to be the best learning rate value.

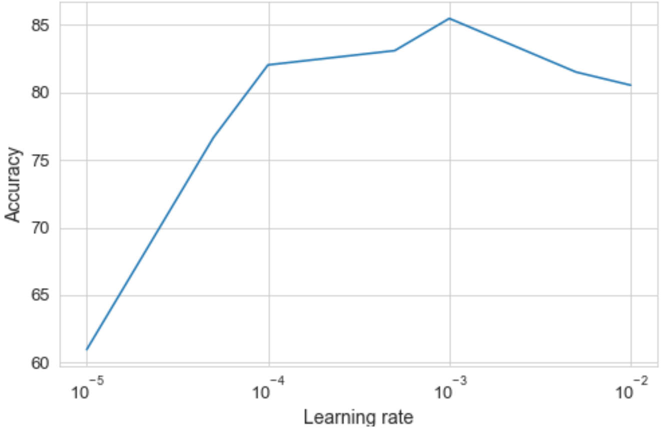

Figure 2: Model accuracy on the dev set per learning rate value

Finally I tune the number of epochs. For these models I again use 768 hidden layers, and a learning rate of $1e^{-3}$.

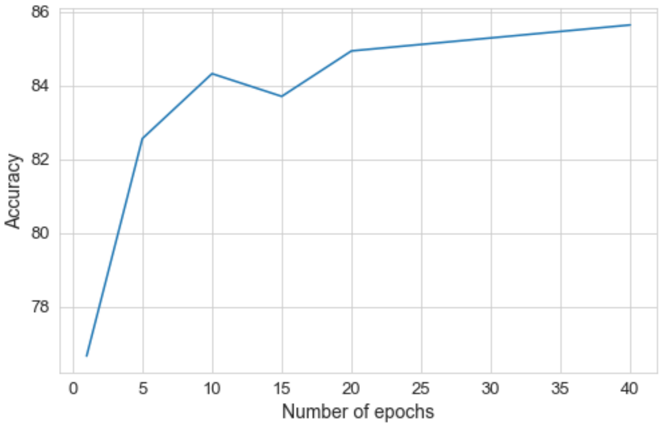

Figure 3: Model accuracy on the dev set per number of epochs

The results can be found in Figure 3, which shows that increasing the number of epochs quite consistently improves the performance (except for a small accuracy drop for 15 epochs). The best accuracy score - of 85.6 - is achieved by the model that is trained for 40 epochs. This model is also best in terms of F1-score (87.0). It achieves a precision score of 80.5 and a recall of 94.6. The best precision score (82.0) is found when training for only one epoch, and the best recall is reached (94.8) after training for 30 epochs. But as training for 40 epochs gives the best accuracy and the best precision/recall trade-off, I consider this the best value.

I therefore conclude the best results are found with 40 epochs, a learning rate of $1e^{-3}$ and 768 hidden layers.

Experiment 3 Finally, I compare the performance of the best Neural Network on the test set to the two baseline models. The results are shown in Table 2. The NN model outperforms the SVM on all metrics except precision, improving the accuracy by 7.5 points. Compared to the Logistic Regression model, the NN performs better on all metrics. In terms of accuracy, the NN is an improvement of 6.6 over this model. I thus consider the NN to be the best model.

Method	**Accuracy**	**Precision**	**Recall**	**F1-Score**
Logistic Regression	76.3	70.0	92.1	79.5
Support Vector Machine	75.4	**86.0**	71.0	77.8
Neural Network	**82.9**	77.0	**93.3**	**84.6**

Table 2: Peformance scores on the test set for the best Neural Network and the two baseline models: Logistic Regression and Support Vector Machine

5 Error analysis

I inspect the false positive and the false negative sentence predictions for the best model (the NN) in the test set. The test set consists of 228 sentences, of which 114 are labelled as sexist, and 114 are labelled as neutral. Of the sexist sentences, the model incorrectly classifies 8 as neutral (false negatives); and it classifies 32 of the neutral sentences as containing sexism (false positives). There are thus more false positives than there are false negatives, which is also reflected in the fact that the recall for this model is higher than the precision. A complete overview of all false positive and false negative sentences in the test set can be found in the Appendix.

An example of a false negative is the following sentence:

- *you can thank the quota*

Here the sexism is very implicit: it requires the knowledge that *quota* refers to the specific quota of having a certain number of female employees (on certain positions) in a company. But as quota can exist for a variety of different things, the fact that this sentence is considered to be sexist is context dependent. It therefore makes sense that this sentence is hard to classify. Another example of a false negative, in which the sexism is more explicit, is the following sentence:

- *for a woman , that is good*

This utterance is clearly sexist, as the speaker expresses to have a different standard of quality for women than for men. A potential reason why the model might consider this sentence to be neutral, is because on a word level, the expression appears to be positive (containing the word *good*), while it is only on the sentence level that the sexism becomes apparent.

An example of a false positive is the sentence below:

- *woman are n't created weaker than men . without woman , everyone 's nothing .*

This sentence contains several negations (*n't, without, nothing*). Potentially here the presence of words like *weaker than men* indicates a strong signal of sexism, and the fact that this was actually negated by the speaker was not picked up on by the model.

6 Discussion

In the experiments I have trained a Neural Network that shows a decent performance - an accuracy of 82.9 and an F1-score of 84.6 - on the task of automated sexism recognition. Grosz & Conde-Cespedes (2020) find an F-score of 88.0 using a BiLSTM model (they do not report the accuracy) - showing that even better results can be found on this dataset. Still, while the Neural Network I use is less complex, the model of Grosz & Conde-Cespedes (2020) only improves the performance by 3.4 points in terms of F1-score. A further experiment to carry out using a Neural Network on this dataset in the future would be to experiment with adding more than one hidden layer. Another possibility to try in order to further improve the performance could be to use a more sophisticated method to create sentence embeddings, such as using a Convolutional Neural Network. Additionally, a State-of-the-Art model such as BERT (Devlin et al., 2018) could be trained on this dataset, which might perform better than the BiLSTM by Grosz & Conde-Cespedes (2020).

A limitation of the current study is that the Sexist Workplace Statements Dataset is rather balanced, while this does not reflect the real-world situtation in which most sentences will not express sexism. This creates a risk when applying the current model in a context where the number of sexist sentences is much lower, as the model might not perform as well. To overcome this, it would be interesting to experiment with adding more non-sexist sentences to the data, and to compare whether a good performance can still be achieved.

Continuing, the sentences in the current dataset fall inside a very specific domain. However, as sentences at the workplace are usually spoken and not written, directly applying the current model at the workplace would be complicated. It would be more realistic to apply such a model in a related, but slightly different domain, for instance to work-related emails or work-related reports and documents. Here, the context would be similar, but the tone and style would be different. For this reason, it could be interesting to carry out domain adaptation experiments, to see how models trained on the current dataset perform on a domain that slightly differs.

Finally, in future work experiments could be carried out with creating an explainable sexism detection model. While the current model can indicate whether a sentence contains sexism, it does not provide any explanation of *why* it classifies the sentence this way. For the application of a sexism detection model, providing such an explanation could allow users to decide whether they trust the model (Rudin et al., 2022; Ribeiro et al., 2016) and could provide speakers with a reason why their utterances might be harmful. Moreover, if decisions are made based on noise rather than on true indicators of sexism (as e.g. in (Ribeiro et al., 2016)), this would also become visible.

7 Conclusion

In this study I have investigated the task of automated sexism recognition using Neural Networks. I have compared five methods of creating semantic sentence vector representations, and found that taking the mean of GloVe embeddings worked the best. Continuing, I tuned three hyperparameters and found 768 hidden layers, a learning rate of $1e^{-3}$ and training for 40 epochs to work best. Finally I found this best Neural network to achieve an accuracy of 82.9, which was an improvement over both baseline models.

Future work could further extend these experiments, e.g. by varying the number of hidden layers or using Convolutional Neural Network to create sentence representations. Further, in order to apply a sexism detection model in a real-world situation, more neutral sentences could be added to the dataset.

References

Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with python: analyzing text with the natural language toolkit.* " O'Reilly Media, Inc.".

De Gibert, O., Perez, N., García-Pablos, A., & Cuadros, M. (2018). Hate speech dataset from a white supremacy forum. *arXiv preprint arXiv:1809.04444*.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Glick, P., & Fiske, S. T. (1996). The ambivalent sexism inventory: Differentiating hostile and benevolent sexism. *Journal of personality and social psychology*, *70*(3), 491.

Grosz, D., & Conde-Cespedes, P. (2020). Automatic detection of sexist statements commonly used at the workplace. In *Pacific-asia conference on knowledge discovery and data mining* (pp. 104–115).

Jha, A., & Mamidi, R. (2017). When does a compliment become sexist? analysis and classification of ambivalent sexism using twitter data. In *Proceedings of the second workshop on nlp and computational social science* (pp. 7–16).

Krivkovich, A., Robinson, K., Starikova, I., Valentino, R., & Yee, L. (2017). Women in the workplace. *McKinsey & Company*. Retrieved from https://www.mckinsey.com/featuredinsights/gender-equality/women-in-the-workplace-2017

Luhn, H. P. (1957). A statistical approach to mechanized encoding and searching of literary information. *IBM Journal of research and development*, *1*(4), 309–317.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, *26*.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (emnlp)* (pp. 1532–1543).

Qian, J., Bethke, A., Liu, Y., Belding, E., & Wang, W. Y. (2019). A benchmark dataset for learning to intervene in online hate speech. *arXiv preprint arXiv:1909.04251*.

Rashid, T. (2016). *Make your own neural network: a gentle journey through the mathematics of neural networks, and making your own using the python computer language.* CreateSpace Independent Publishing Platform.

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 1135–1144).

Rudin, C., Chen, C., Chen, Z., Huang, H., Semenova, L., & Zhong, C. (2022). Interpretable machine learning: Fundamental principles and 10 grand challenges. *Statistics Surveys*, *16*, 1–85.

Samory, M., Sen, I., Kohne, J., Flöck, F., & Wagner, C. (2021). Call me sexist, but...: Revisiting sexism detection using psychological scales and adversarial samples. In *Intl aaai conf. web and social media* (pp. 573–584).

Waseem, Z., & Hovy, D. (2016). Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In *Proceedings of the naacl student research workshop* (pp. 88–93).

A Appendix

All false negative sentences:
test are like woman . hard to understand , easy to get over , and nonrefundable .
i just think men are better comedian .
what will we do without feminist ? ! everything .
for a woman , that is good
you can thank the quota
how do you sleep at night , cathy ? you . cunt .
for me , she is not strong enough
i think in general , men make way better music

All false positive sentences:
i hate being placed on committee . they are always having meeting at which half are absent and the rest late .
i never sleep well when i 'm on call .
you are a intelligent a this guy
do you and your little girl watch sport together ?
the woman 's party is made up of woman of all race , creed and nationality who are united on the one program of working to raise the status of woman .
i 'm not someone who 's led my life trying to get publicity i 'd rather do my work and go home .
all job are odd , or they would be game or nap or picnic .
when a man is ambitious it 's seen a a good thing , and when a woman is described a ambitious it 's seen a a complete negative
people buy into the leader before they buy into the vision
five foot and nine inch is tall for a girl right ?
a woman is just a good a a man
i 'm just a good a a man . i do n't need a special day to celebrate that .
men can have parental leave a woman
most business meeting involve one party elaborately suppressing a wish to shout at the other : 'just give u the money ' .
you get a promotion
if mainstream culture think gender role are unimportant , church culture make them too important .
human are men and woman
my frustration with kat is getting stronger and stronger every time i meet with her and her team .
the president notice that when he take off his coat to dig , people take more notice of the visual than they did his preceding remark .
you are a professional woman
what would the world do without woman and girl ?
that wa in reply to her .
and i 'm very lucky to have a girl in my life who i have the privilege to call my wife
woman first !
when life give you lemon squirt them into the eye of people who tell you that you are smart for a girl rape is not a

punchline .
my mind feel like a beehive without the buzz .
woman are n't created weaker than men . without woman , everyone 's nothing .
woman can do anything a man can
let others slap each others on the back while you 're back in the lab or the gym or pounding the pavement .
do n't think that a woman ha to act like a man to show that she ha strength .
if you are lazy , then undoubtedly you will soon become poor
managing is getting paid for home run someone else hit .