194.093 NLP and IE — Milestone 2
Group 21

Sven Gerloff - 52303639
Theresa Mayer - 11919716
Yasmine Khajjou - 12409115
Thomas Klar - 12021340

10.11.2024

Contents

1 Introduction

In the second milestone, our goal is to deploy various baseline models, including both deep learning and non-deep learning approaches. The focus is on establishing an experimental setup that allows further experimentation, such as testing different datasets, pre-trained models, and hyperparameters, while tracking of results throughout the project. The performance of the baseline models will be analyzed both quantitatively and qualitatively to identify potential issues that can be addressed in the next milestone.

2 Experiment Setup

We are using Weights and Biases (WandB) to track and analyze our experiments [1]. The goal was to implement a script for each model to track specific hyperparameter settings and performance measurements. We created a project in WandB that can be accessed and used by everyone, providing a single source of truth for our experiments. Figure 1 shows an example of tracking experiments for the deep learning model.

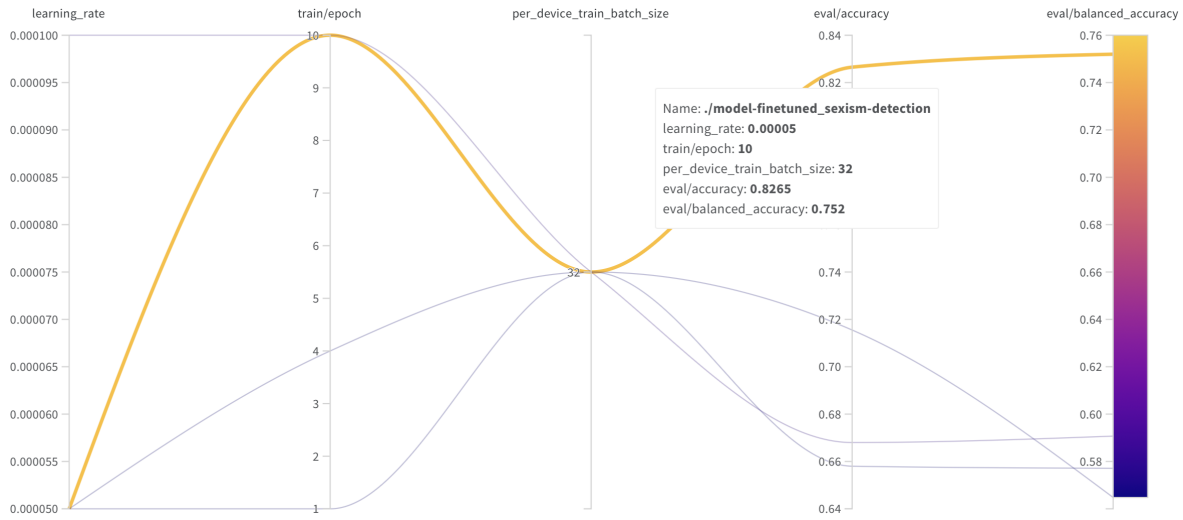

Figure 1: Deep Learning Experiments with WandB

3 Baseline Models

3.1 Non-Deep Learning

3.1.1 Naive Bayes

Our first non-deep learning model is Multinomial Naive Bayes, which is used for text classification [3]. We chose it because it is a simple model, which was also discussed in the lecture. An advantage of the model is that it provides probabilities for both classes, allowing for a more detailed analysis of the results.

As input for the model, we used the prepared dataset from milestone 1, where we applied different steps for data cleaning and lemmatization. In addition, we used the TfidfVectorizer to further prepare the data and focus on the most important and valuable terms [4]. For TF-IDF, we conducted different experiments and logged the results in Weights and Biases. Table 1 summarizes all the parameters we tried. The best-performing configuration is highlighted in bold.

Table 1: Naive Bayes - TF-IDF

max_features	ngram_range	min_df
3000	(1, 1)	5
3000	(1, 2)	5
5000	(1, 1)	5
5000	**(1, 2)**	**5**
10000	(1, 1)	5
10000	(1, 2)	5

3.1.2 Logistic Regression

Our second on-deep learning model is Logistic Regression, a widely used supervised learning algorithm for binary classification tasks. We chose it because of its simplicity and interpretability, making it a good baseline for comparison with other models. An advantage of Logistic Regression is that it provides probabilities for both classes, allowing for a more detailed analysis of the results.

As input for the model, we used the prepared dataset from milestone 1, where we applied the same steps in Naive Bayes for data cleaning and lemmatization. In addition, we used two feature extraction techniques: TF-IDF and Word2Vec. We conducted experiments with other parameters for TF-IDF, including max features, n-gram range, and minimum document frequency. The parameters we tried are summarized in Table 1. We also experimented with different Word2Vec parameters, including vector size, window size, and minimum word count. The parameters we tried are summarized in Table 2.

We also used the SMOTE oversampling technique to handle class imbalance in the dataset. We then trained a Logistic Regression model with balanced class weights and evaluated its performance using accuracy, balanced accuracy, and F1 score. We logged the results in Weights and Biases for TF-IDF and Word2Vec experiments.

Table 2: TF-IDF Parameters

Max Features	N-gram Range	Min DF
3000	(1, 1)	5
3000	(1, 2)	5
5000	(1, 1)	5
5000	(1, 2)	5
10000	(1, 1)	5
10000	(1, 2)	5

Table 3: Word2Vec Parameters

Vector Size	Window Size	Min Count
50	5	1
100	5	1
200	5	1
300	5	1
300	10	1
300	20	1

3.1.3 Support Vector Machine

Our last non-deep learning model is Support Vector Machine (SVM), a widely used supervised learning algorithm for binary classification tasks. It was selected because of its ability to handle high-dimensional data and its robustness to noise.

As input for the model, we used the prepared dataset from milestone 1. In addition, we used the TF-IDF feature extraction technique to further prepare the data and focus on the most important and valuable terms. For TF-IDF, we conducted experiments with different parameters, including max features, n-gram range, and minimum document frequency. The parameters we tried are summarized in Table 4.

We used the SMOTE oversampling technique to handle class imbalance in the dataset. We then trained an SVM model with balanced class weights and evaluated its performance using accuracy, balanced accuracy, and F1 score. We logged the results in Weights and Biases for each TF-IDF configuration.

Table 4: TF-IDF Parameters for SVM

Max Features	**N-gram Range**	**Min DF**
3000	(1, 1)	5
3000	(1, 2)	5
5000	(1, 1)	5
5000	(1, 2)	5
10000	(1, 1)	5
10000	(1, 2)	5

3.2 Deep Learning

We decided to use a pre-trained transformer model from huggingface.com and finetune it on our labeled training data. The model we picked is called the *Toxic-Comment-model* [2] by user martin-ha. This model is based on the DistilBERT architecture [5]. It was originally intended to be used to classify toxic comments in any online space. Since we suspect that toxic comments share a lot of characteristics with sexist comments (or even are a superset of them) and this is a model trained on short texts of typical netspeak, we believe this model to be very suitable for our task.

We further trained a classifier with our training data on this model. As mentioned in the previous section, we used Weights and Biases to track performance metrics and optimize hyperparameters. However, due to the long training times of the model, we could not devote a lot of time to this optimization, and it was also not the focus of this milestone.

The model uses a tokenizer to transform the input data, so we could not use our preprocessed data from milestone 1 here. Instead, we tokenized the raw input data since this is more in line with the functionality of the tokenizer, which does the required preprocessing internally.

Additionally, we determined the optimal cutoff value by investigating the ROC-AUC-curve and choosing accordingly. In most experiments, we observed low cutoffs at around 0.25. This means the classifier will output class 1 (Sexist) even if the predicted probability

of a comment being sexist is equal or greater to 25%. This tends to improve recall but decreases precision.

4 Results

4.1 Quantitative

The performance of the various models used for the online sexism detection task is summarized in Tables 5, 6, 7, 8, and 9.

Table 5: Classification Report for Logistic Regression using TF-IDF on the Test Set

Class	**Precision**	**Recall**	**F1-Score**	**Support**
0 (Non-Sexist)	0.89	0.85	0.87	3030
1 (Sexist)	0.59	0.66	0.62	970
Accuracy	0.81			
Balanced Accuracy	0.76			
Macro Avg	0.74	0.76	0.75	4000
Weighted Avg	0.81	0.81	0.81	4000

Table 6: Classification Report for Logistic Regression with Word2Vec on the Test Set

Class	**Precision**	**Recall**	**F1-Score**	**Support**
0 (Non-Sexist)	0.82	0.53	0.65	3030
1 (Sexist)	0.30	0.63	0.41	970
Accuracy	0.56			
Balanced Accuracy	0.58			
Macro Avg	0.56	0.58	0.53	4000
Weighted Avg	0.69	0.56	0.59	4000

Table 7: Classification Report for SVM using TF-IDF on the Test Set

Class	**Precision**	**Recall**	**F1-Score**	**Support**
0 (Non-Sexist)	0.83	0.97	0.89	3030
1 (Sexist)	0.82	0.36	0.50	970
Accuracy	0.82			
Balanced Accuracy	0.67			
Macro Avg	0.82	0.67	0.70	4000
Weighted Avg	0.82	0.82	0.80	4000

Table 8: Classification Report for Naive Bayes on the Test Set

Class	**Precision**	**Recall**	**F1-Score**	**Support**
0 (Non-Sexist)	0.79	0.99	0.88	3030
1 (Sexist)	0.92	0.18	0.30	970
Accuracy	0.80			
Balanced Accuracy	0.59			
Macro Avg	0.85	0.59	0.59	4000
Weighted Avg	0.82	0.80	0.74	4000

The Deep Learning Classifier achieves about the same scores on recall and precision for both classes individually. In addition, the accuracy and the balanced accuracy are comparable. Precision and recall are higher for Non-Sexist samples. This suggests that this classifier has difficulties with this class, and further improvements are necessary.

Table 9: Classification Report for DL Classifier on the Test Set

Class	**Precision**	**Recall**	**F1-Score**	**Support**
0 (Non-Sexist)	0.89	0.88	0.89	3030
1 (Sexist)	0.64	0.66	0.65	970
Accuracy	0.83			
Balanced Accuracy	0.77			
Macro Avg	0.77	0.77	0.77	4000
Weighted Avg	0.83	0.83	0.83	4000

Table 10: Classifiers Performance Metrics on the Test Set

Classifier	**Accuracy**	**Balanced Accuracy**	**F1 Weighted**	**F1 Macro**
Logistic Regression (TF-IDF)	0.81	0.75	0.81	0.75
Logistic Regression (Word2Vec)	0.57	0.60	0.60	0.54
Naive Bayes	0.80	0.59	0.74	0.59
DL Transformer	0.83	0.77	0.83	0.77

4.2 Qualitative

In order to get a good overview of the incorrect predictions, it was first examined how many models incorrectly classified each text. It emerged that over 1600 texts were correctly recognized by all classifiers. Approximately 1400 texts were incorrectly classified by one classifier, around 500 by two and only about 150 by all four.

To get an even better insight, a correlation matrix was created to show the correlation between the different models for the predictions.

For the qualitative analysis, the focus was now primarily on those examples that were incorrectly classified by all models. Reading through the texts, it became apparent that some of them were labeled incorrectly. For example: "Why did he move out? She's the cheater, she should be forced out." was labeled as sexist, while we would rather label it as not sexist, as the 4 models predicted. Another striking feature is that some comments provide no context and could maybe therefore be hard to categorize purely on the text.

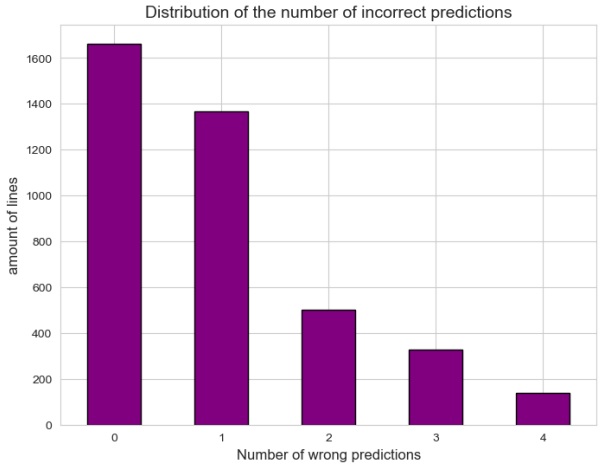

Figure 2: Number of models which made a wrong prediction

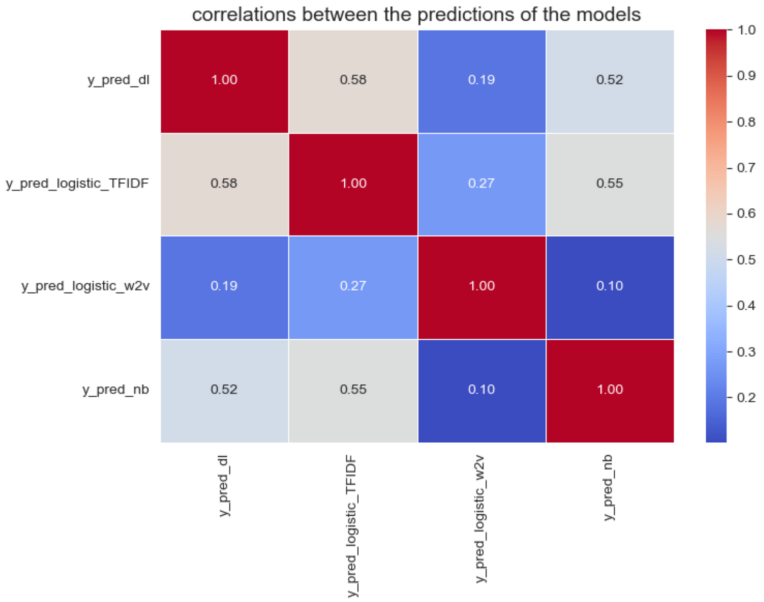

Figure 3: Correlation between the predictions of the different models

An example would be: "Well said. Single mothers would not get this point." which is originally labeled as sexist. Another noticeable aspect of some texts is that there were misspellings or random characters included. This could be a significant issue in general. This can be seen for example in this case: "If youâ€™re 6â€™3â€ chad with a 2â€ dick she will find another chad. Maybe not an incel but fuck I can sympathize.,1".

5 Possible Sources of Malfunction and Solution

The EDOS dataset[6] includes 14,000 labeled training samples. This relatively small dataset may pose challenges, especially as tasks become more complex or when models require additional data to improve performance. Within the training set, there is an imbalanced class distribution: 3,398 instances labeled as "sexist" and 10,602 labeled as "not sexist". This imbalance is evident in the classification reports generated using Scikit-learn, where the performance metrics for the "not sexist" class are consistently higher than for the "sexist" class, with a noticeable gap (e.g. Table 8). This discrepancy is also reflected in the macro-average metrics, likely influenced by both the limited dataset size and the class imbalance.

To address these issues, we will employ various data augmentation strategies to expand the dataset and balance the class distribution in the next milestone. As illustrated in Figure 4, different methods for data augmentation in NLP are presented. Data Augmentation can also improve the generalization and robustness [7]. If we are not successful with that solution, another possible approach would be to use a different dataset with the same or a similar task, pre-train on that dataset, and then fine-tune with the EDOS dataset.

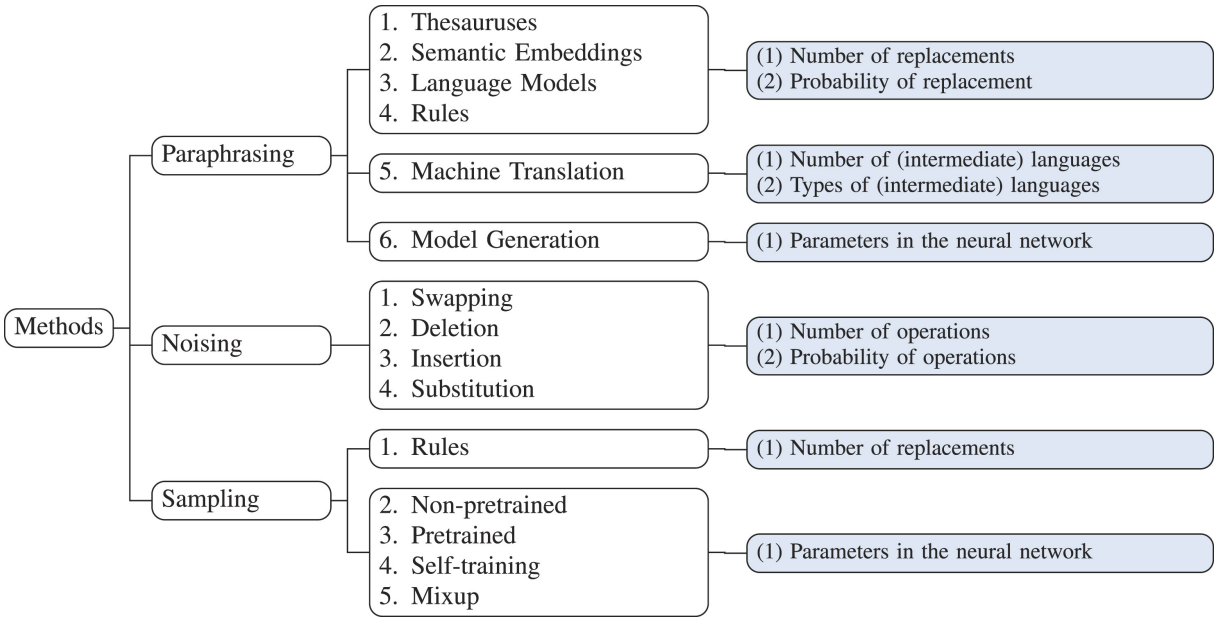

Figure 4: Data Augmentation Methods [7]

References

[1] Weights and Biases, https://wandb.ai.

[2] Toxic Comment model, https://huggingface.co/martin-ha/toxic-comment-model

[3] Multinomial Naive Bayes, https://scikit-learn.org/1.5/modules/naive_bayes.html#multinomial-naive-bayes

[4] TfidfVectorizer, https://scikit-learn.org/1.5/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

[5] Sanh, Victor, Lysandre Debut, Julien Chaumond and Thomas Wolf. *DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter.* https://arxiv.org/abs/1910.01108

[6] Rewire Online. *EDOS Dataset (Labelled Aggregated Data).* Retrieved from https://github.com/rewire-online/edos/blob/main/data/edos_labelled_aggregated.csv.

[7] Bohan Li, Yutai Hou, Wanxiang Che. *Data augmentation approaches in natural language processing: A survey.* AI Open, Volume 3, 2022, Pages 71-90. ISSN 2666-6510. Retrieved from https://doi.org/10.1016/j.aiopen.2022.03.001. (Accessed via: https://www.sciencedirect.com/science/article/pii/S2666651022000080).