

---

# Sentiment Analysis

## CIL 2025

---

Besche Awdir<sup>1</sup> Edi Zeqiri<sup>1</sup> Mario Tachikawa<sup>1</sup> Hamza Zarah<sup>1</sup>

### Abstract

This paper presents a transformer-based approach for sentiment analysis of restaurant reviews, classifying text into positive, neutral, or negative categories. We fine-tune DeBERTa-v3 using a dataset of approximately 102,000 labeled reviews, implementing hyper-parameter search and Easy Data Augmentation (EDA) to expand the training data. Our method outperforms classical (Logistic Regression + BoW) and transformer baselines (DistilBERT, Twitter-RoBERTa), achieving a custom evaluation score of 0.8943 (MAE: 0.2114) on our validation set, through disentangled attention mechanisms and hyperparameter optimization.

## 1. Introduction

### 1.1. Scope and Objectives

The primary objective is to develop a sentiment classification model using the provided dataset of approximate 102'000 labeled reviews and 12'000 test samples. Key goals include:

- Implementing preprocessing strategies to handle noise (e.g., URLs, typos, repeated characters).
- Fine-tuning the roBERTa transformer model to leverage its contextual understanding capabilities.
- Optimizing performance using the evaluation metric  $L(\hat{y}, y) = 0.5 \cdot (2 - \text{MAE}(\hat{y}, y))$ , which penalizes severe misclassifications more heavily.

### 1.2. Organization of the Paper

The remainder of this report is organized as follows. Section *background* reviews theoretical and computational work on semantics and sentiment analysis. Section *Methods* introduces which methods we tried out as our baseline and effectively which model we used to get the highest score. Section *Experiments & Results* details the evaluation of our models, highlighting the comparisons and achieved scores. Finally, Section *Discussion & Conclusion* discusses the results, strengths, limitations, future work and summarizes the report.

## 2. Background and Related Work

### 2.1. Fundamentals of Semantics

Sentiment hinges on two layers of meaning: **lexical polarity**, the default valence of single words (*great*, *terrible*), and **compositional effects**, where syntax or function words flip/scale that valence (*not great*, *very good*). Early work quantified polarity as *semantic orientation*—a word's co-occurrence bias toward *excellent* vs. *poor* (Turney, 2002). Modern neural encoders learn both layers jointly, yielding context-sensitive embeddings without explicit rules.

### 2.2. Linguistic Theories of Meaning

Linguistic theory underscores three challenges that any sentiment model must solve:

1. *Polarity lexicons* from lexical semantics can bootstrap priors but ignore context;
2. *Compositionality* requires rules (or learned functions) that flip or scale sentiment under negation, intensification, or concessives;
3. *Pragmatics* complicates matters via sarcasm and implicature—cases where literal words mislead. Rule-based systems hard-code (i) and a few patterns for (ii); statistical and neural models aim to learn all three directly from data.

### 2.3. Dataset overview

We were provided with two comma-separated files: `training.csv` with 102 097 labelled sentences and `test.csv` with 11 951 unlabelled sentences.<sup>1</sup> For model development we create a stratified 90:10 partition of the training corpus, referred to throughout as *training\_split* and *validation\_split*.

**Class balance.** Table 1 reveals a moderate imbalance: almost one review in two is *neutral*, whereas *positive* and *negative* together form the remaining half.

<sup>1</sup>Identifiers are unique only *within* a split; an `id=42` in the test set bears no relation to `id=42` in the training set.

Sentiment	Count	Share (%)
neutral	49 148	48.1
positive	31 039	30.4
negative	21 910	21.5

Table 1. Label distribution in the official training corpus.

## 2.4. Data Augmentation

To enlarge the training dataset without using any external datasets we applied data augmentation to the given data. With *Easy Data Augmentation* (EDA) (Wei & Zou, 2019) we augmented additional 227'180 data points to our training dataset using the four methods described in the paper (Wei & Zou, 2019): Synonym Replacement (SR), Random Insertion (RI), Random Swap (RS), Random Deletion (RD).

We set  $\alpha = 0.1$  and  $n_{\text{aug}} = 3$ , as suggested in the paper, meaning that for every data-sentence 3 additional variants are added where 10% of the words were changed or deleted. In the implementation, WordNet-based synonyms and an NLTK stop-word filter were used.

The goal of this process is to enlarge the training dataset and to add diversity, such that the model generalizes more.

## 3. Methods

### 3.1. Classical baseline: LogReg + BoW

To have a reference to all the transformer models which we used and will describe later, a traditional bag-of-words logistic regression model was implemented serving as the simple, classical baseline.

We tokenized the sentences and build a bag-of-words representation for the training data. Using the CountVectorizer, we built a vocabulary of the 10'000 most frequent uni- and bi-grams, and converted every training sentence into a vector of token counts. After L2-normalization, a one-vs-rest logistic regression is trained using the default LBFGS solver (L2 penalty,  $C = 1.0$ , max-iter = 300). The model then predicts the labels of the validation set using softmax on the three sentiment logits and taking the argmax as the prediction. (The implementation is based on the lectures introduction notebook on sentiment analysis)

### 3.2. Transformer baselines

All transformer runs share the same pre-processing and optimization process. The review texts are tokenized with the AutoTokenizer of the corresponding model. The sentences are then truncated or padded to 128 maximum length. The model then predicts the labels using a three way classification head (num\_labels = 3), which returns logits for each sentiment label and using softmax the highest probability class is predicted. During training, the cross-entropy loss

between the predicted label and the true label is minimized. The prediction during evaluation time is the argmax of the soft-max probabilities, and metrics are computed on this predicted label. Fine-tuning the different models using the training split of the data is done on one NVIDIA A100 40GB in Google Colab. Every half epoch we evaluated the model on the validation split. The checkpoint for the best model was set based on the validation MAE. All training runs have set a fixed seed: 42.

#### 3.2.1. Baseline: DistilBERT-base (Sanh et al., 2020)

As our lightweight transformer baseline with the default hyper-parameters we chose DistilBERT-base-cased with about 66M parameters. We fine-tuned the model on the training split using the following hyper-parameters for 3 epochs:

- lr:  $5 \cdot 10^{-5}$
- batch-size: 16
- weight-decay: 0.01
- no warm-up
- FP16 mixed precision

#### 3.3. Intermediate Model: Twitter RoBERTa Sentiment (Barbieri et al., 2020)

As a midway point between our DistilBERT baseline and the fully fine-tuned DeBERTa-v3-base, we employ a RoBERTa checkpoint pre-trained on a large Twitter sentiment corpus with about 125 M parameters. This “twitter-roberta-base-sentiment” model already captures social-media vernacular—user mentions, hashtags, emojis and URL patterns—so we apply only minimal text normalization (lowercasing, user/URL masking, whitespace collapse) before fine-tuning.

The model was fine tuned for 3 epochs using the following hyperparameters:

- learning-rate  $2.0 \cdot 10^{-5}$ ,
- lr-scheduler: cosine,
- warm-up ratio: 0.10
- weight-decay: 0.03
- batch-size: 8,
- FP16 mixed precision

#### 3.3.1. Proposed Model: DeBERTa-v3-base (He et al., 2021)

Our main system uses DeBERTa-v3-base, which extends the BERT architecture with disentangled attention and enhanced position embeddings with about 141 M parameters. We initialize from the publicly available checkpoint and fine-tune all parameters on our task-specific dataset. By combining DeBERTa’s stronger contextual modeling with targeted supervision, we aim to improve the semantic under-

standing of the models that the baseline may miss.

Unlike the purely “default” DistilBERT run, we performed a small random search (30 trials) and retained the configuration with the lowest validation MAE. This procedure is described in the section about hyper parameter tuning 3.4.

We trained the model for 4 epochs using the following hyper-parameters:

- learning-rate  $2.0 \cdot 10^{-5}$ ,
- lr-scheduler: cosine,
- warm-up ratio: 0.10
- weight-decay: 0.02
- batch-size: 8,
- FP16 mixed precision

### 3.4. Hyper-parameter tuning of DeBERTa

To maximize the performance of our DeBERTa model, we made a two stage random search of hyper-parameters that have the best performance on the validation set.

Before searching the best hyper parameter configuration we defined the search spaces on all parameters individually. Then randomly chose 30 configurations which are in the search spaces of all parameters and fine-tuned our model using 10% of the training data (to reduce computational duration) for 1 epoch. These configurations were then ranked by their corresponding MAE values based on the full validation split.

The best 5 configurations were then again used for fine tuning the model but this time for 2 epochs, to analyze their individual performance for a longer training period. The “winner” of these 5 candidates was then used for the fine tuning of the DeBERTa-v3-base model.

Hyper-parameter	Range	Winning value
Learning-rate	$(5 \cdot 10^{-6}, 3 \cdot 10^{-5})$	$2.02 \cdot 10^{-5}$
Weight-decay	$(0, 0.05)$	0.0198
Batch size	{8, 16}	8
LR scheduler	{cosine, linear}	cosine
Warm-up ratio	$(0, 0.20)$	0.10

Table 2. Random-search space and best hyper-parameter set for DeBERTa-v3-base.

#### 3.4.1. Easy Data Augmentation (EDA)

As described in in the section about data-augmentation 2.4, we augmented our data to enlarge the training data used for fine-tuning. The DeBERTa-v3-base model was fine tuned on the non-augmented training split and the augmented training data using the exact same hyperparameters.

### 3.5. Base Architecture

All transformer models share the transformer encoder backbone:

- Embedding layer: Token, position (and for DeBERTa disentangled) embeddings.
- Multi-head self-attention: Captures inter-token dependencies across layers.
- Feed-forward network: Two-layer MLP with GELU activation.
- Classification head: A linear layer on top of the token, producing logits.

## 4. Experiments & Results

We evaluated our classifiers described in 3 with increasing complexity on our sentiment-analysis benchmark.

### 4.1. Experimental set-up

All our models were evaluated using the same validation split (10% of the training set). Our transformer models are evaluated on the validation dataset using three different seeds {13, 21, 42}. Since the logistic regression model is deterministic after training, we report one single inference run. All inference was done using one NVIDIA A100 (40GB) GPU in FP16. The metrics are reported as the mean +/- standard deviation over the three inference runs with different seeds.

Such that the seeds have an impact on the transformer models, we implemented a Monte-Carlo dropout sampling of size 3 per seed. This can be done by leaving the trainer of the model in training mode (rather than evaluation mode). Each sentence leads to one mean probability vector:

$$\bar{p} = \frac{1}{3} \sum_{m=1}^3 p^{(m)}$$

Using this mean probability vector we predict the label of the sentence using argmax:

$$y_{\text{pred}} = \text{argmax}(\bar{p})$$

Using these predictions we calculate the MAE and the evaluation score (L-score):

$$L(\hat{y}, y) = 0.5 \left( 2 - \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \right),$$

### 4.2. Overall Model Comparison

The following table summarizes the evaluation of the models. Scores were averaged over the three evaluation seeds (except for LogReg). The values are displayed as *mean ± standard deviation*.

Model	L-score $\uparrow$	MAE $\downarrow$
LogReg + BoW	0.7800	0.4402
DistilBERT	$0.8437 \pm 0.0004$	$0.3127 \pm 0.0008$
Twitter-RoBERTa	$0.8699 \pm 0.0001$	$0.2593 \pm 0.0003$
DeBERTa-v3	$0.8899 \pm 0.0010$	$0.2203 \pm 0.0020$
DeBERTa-v3 (+ EDA)	<b><math>0.8943 \pm 0.0002</math></b>	<b><math>0.2114 \pm 0.0004</math></b>

Table 3. Validation performance of all systems.

### Key observations

- classical baseline in comparison to transformers:  
Even the lightweight transformer baseline lowers MAE by 29 % (3) in comparison with the BoW logistic regression.
- transformer comparisons:  
Twitter-RoBERTa (125 M parameters) outperforms DistilBERT, and the strongest DeBERTa-v3 (141 M) pushes MAE down a further 15.1 %.
- Stability:  
Standard deviations are  $\leq 0.002$  for all Transformer runs.

### 4.3. Effect of Data Augmentation

To show the difference of the DeBERTa-v3 model using the given training dataset and the data augmented dataset described in 4.3, we isolated the evaluation results of the two inference runs:

Variant	MAE $\downarrow$	$\Delta$
DeBERTa-v3	$0.2203 \pm 0.0020$	–
DeBERTa-v3 (+ EDA)	<b><math>0.2114 \pm 0.0004</math></b>	$-0.0089$ (4.0 %)

Table 4. Impact of EDA on DeBERTa-v3.

We see that the MAE drops by 4% when using data augmentation. We can therefore conclude, that having a larger training set and adding diversity to the data, improves the models MAE in comparison to using the smaller, given training set.

### 4.4. Per-class performance and error patterns

Detailed, confusion matrices for all transformer models for the predictions on the validation set are provided in Appendix A (Fig. 1–4). The following observations are clear:

- (a) DistilBERT’s main weakness is that false classification of positive and negative sentences as neutral sentences. For both directions, the model falsely predicts about a quarter of the sentences as neutral, even though their true labels are non-neutral.

- (b) Moving from DistilBERT to Twitter-RoBERTa and further to DeBERTa increases the percentages for the true positives for all three classes. We can also see that our final proposal model has the best values in all cases, meaning that the models performance is stable throughout all sentiment classes.

## 5. Discussion & Conclusion

**Model selection:** The step from the classical BoW + LogReg baseline to even the classical transformer (DistilBERT) baseline removes nearly one-third of the average absolute error (3). The stronger transformer models pre-trained on in-domain text continue this trend: Twitter-RoBERTa trims MAE by a further 17% and our strongest candidate DeBERTa-v3 using EDA delivers the best single-model score (0.211 MAE).

**Data volume still pays dividends.** Easy Data Augmentation (EDA) provides a modest but consistent 4 % MAE reduction on top of the fine-tuned DeBERTa-v3 model (Table 4). Because hyper-parameters and seeds are identical between the two runs, we can conclude that this gain comes purely from the enlargement of the training data and diversity of the data produced by the EDA process.

**Remaining problems** The confusion matrices show that all models still struggle with the positive and negative labeled sentences the most. DistilBERT mislabels almost a quarter of the truly positive/negative sentences as neutral, while DeBERTa-v3 reduces this to one-fifth, and almost eliminates “positive  $\leftrightarrow$  negative” misclassifications, but still cannot prove total generalization over all classes.

**Limitations and future work** Our augmentation strategy is rule-based and monolingual. Other data augmentation strategies like back-translation or paraphrase models could have an advantage over our data augmentation process. We also evaluate on a single validation split, where K-fold cross validation and ensemble assessment would remove any bias created by an exact split of the labeled data. In future work these improvements could be analyzed with the goal of improving even more.

**Summary** A careful progression from a classical baseline to stronger transformer models, a random search of hyper-parameter configurations and a data augmentation step, led to the best L-score of **0.88544** (Kaggle public score) and cutting the MAE of the DistilBERT baseline by 32% on the validation set, while maintaining reproducibility through fixed seeds and shared hyper-parameters. Our comparisons of the DeBERTa-v3 transformer model using data augmentation show that this model achieved the best performance in terms of MAE and accuracy in all classes.

## References

- Barbieri, F., Camacho-Collados, J., Espinosa Anke, L., and Neves, L. TweetEval: Unified benchmark and comparative evaluation for tweet classification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 1644–1650, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.148. URL <https://aclanthology.org/2020.findings-emnlp.148>.
- He, P., Gao, J., and Chen, W. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing, 2021.
- Sanh, V., Debut, L., Chaumond, J., and Wolf, T. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, 2020. URL <https://arxiv.org/abs/1910.01108>.
- Wei, J. and Zou, K. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In Inui, K., Jiang, J., Ng, V., and Wan, X. (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 6382–6388, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1670. URL <https://aclanthology.org/D19-1670/>.
- V. Hatzivassiloglou and K. R. McKeown. Predicting the semantic orientation of adjectives. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 174–181, 1997.
- P. D. Turney. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 417–424, 2002.
- B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up? Sentiment classification using machine learning techniques. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 79–86, 2002.
- B. Pang and L. Lee. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1–2):1–135, 2008.
- B. Liu. *Sentiment Analysis and Opinion Mining*. Morgan & Claypool, 2012.
- Y. Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, 2014.
- R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1631–1642, 2013.
- D. Tang, B. Qin, and T. Liu. Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1422–1432, 2015.
- J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. 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*, pages 4171–4186, 2019.
- Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

## A. Appendix: Confusion Matrices

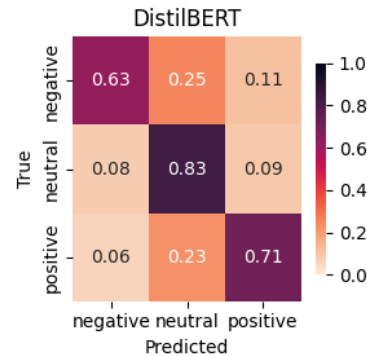


Figure 1. Confusion matrix for DistilBERT on the validation set.

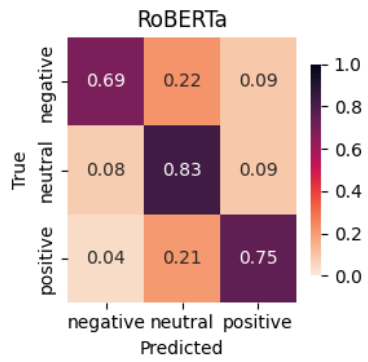


Figure 2. Confusion matrix for RoBERTa on the validation set.

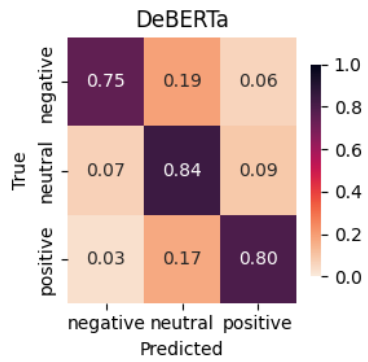


Figure 3. Confusion matrix for DeBERTa-v3 on the validation set.

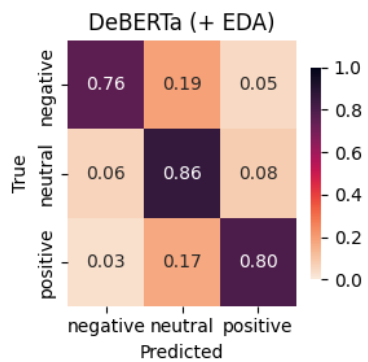


Figure 4. Confusion matrix for DeBERTa-v3 + EDA on the validation set.



Eidgenössische Technische Hochschule Zürich  
Swiss Federal Institute of Technology Zurich

## Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

**Title of work** (in block letters):

Sentiment Analysis

**Authored by** (in block letters):

*For papers written by groups the names of all authors are required.*

**Name(s):**

Tachikawa

Zeqiri

Awdir

Zarah

**First name(s):**

Mario

Edi

Besche

Hamza

With my signature I confirm that

- I have committed none of the forms of plagiarism described in the '[Citation etiquette](#)' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

**Place, date**

Zurich, 29 May 2025

**Signature(s)**

Besche

Edi

Hamza

M. Tachikawa

*For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.*