Entity-Level Sentiment Analysis with Sentence Relevance Detection

Egil Rønningstad[1,2], Roman Klinger[2], Lilja Øvrelid[1], Erik Velldal[1]
[1]Department of Informatics, University of Oslo, Norway
[2]Fundamentals of Natural Language Processing, University of Bamberg, Germany
{egilron, liljao, erikve}@uio.no
roman.klinger@uni-bamberg.de

Abstract

The task of entity-level sentiment analysis (ELSA) is to extract sentiment scores for a given entity (such as person names or organization names) from a text. ELSA is a challenging task and involves processing of longer documents, where several entities may be mentioned with varying importance for the final score aggregation. Fine-tuning encoder-based Transformers (such as BERT) constitutes the state of the art for sentiment predictions, however, these models are still limited by their restricted input lengths. Decoder-only models so far still underperform on the task. We approach the context limitation by learning to extract segments that are relevant for the sentiment prediction for a given entity, without preprocessing by chunking and aggregation. For decoder models, we explore fine-tuning these through supervised fine-tuning and pairwise comparison, a method borrowed from reward modeling for preference optimization. Both methods perform well and set a new standard for the ELSA task. We further show that pairwise classification is faster, simpler, and shows less variance than the more common direct supervision for this task.

Keywords: Text Classification, Sentiment Analysis, Fine-Tuning

1. Introduction

The recent Norwegian dataset for Entity-Level Sentiment Analysis (ELSA, Rønningstad et al., 2024) poses the challenge of providing separate document-level sentiment labels for each entity (person or organization) mentioned in a text. While early sentiment analysis (SA) research aimed at classifying one text with one sentiment label such as Positive or Negative, SA evolved into more fine-grained tasks, such as classifying text with respect to certain aspect categories (Hu and Liu, 2004; Pontiki et al., 2014), or identifying sentiment expressions with their holder and target in each sentence (Wiebe et al., 2005; Barnes et al., 2022). These fine-grained analyses have mostly been performed on shorter texts, such as individual sentences, user reviews or microblog posts. However, in recent research there is a stronger focus on text classification of longer documents, while maintaining the fine-grained SA classification objectives developed for shorter texts (Cai et al., 2024; Luo et al., 2022; Pi et al., 2024).

The ELSA dataset contributes to this shift towards longer texts, consisting of professional reviews from news sources where a majority of the texts exceed the 512 subword token limit common to encoder models. A model for ELSA needs to identify sentiment expressions as well as resolve long-range dependencies for sentiment targets in a document, as a sentiment in one sentence may relate to an entity only mentioned by name several sentences before. Further, as the overall sentiment regarding each entity is requested, the totality of potentially conflicting sentiments expressed through the text needs to be taken into account.

In recent years, the auto-regressive decoder models have surpassed the encoder models in popularity and capability in many tasks within NLP. However, for text classification tasks such as SA, encoder models still remain ahead of decoder models that are an order of magnitude larger (Bucher and Martini, 2024; Saattrup Nielsen et al., 2025). As our main focus in this work is on editorial content, texts frequently exceed 512 tokens, but may fit well within smaller decoder models' context window of 8K or 4K tokens. In order to deal with these longer documents, our experiments investigate techniques to compress longer texts by locating the text segments containing task-specific information needed for classification, thus allowing encoder models to classify the text without further segmentation. We further evaluate the common method of zero-shot or few-shot prompting of decoder models, as well as two fine-tuning regimes commonly used in instruction-tuning of LLMs: supervised fine-tuning (SFT) and preference modeling through pairwise comparison (PC). For PC, a model is trained to choose the best of two alternative texts, or replies to a prompt, based on labeled data for human preferences. In our work we use the PC training regime to fine-tune a model to choose a text containing the true class label over an alternative text containing an incorrect label. While SFT has been established as an approach for text classification tasks, we have not observed PC being used for this purpose. Accordingly, the present work emphasizes PC over SFT.

Our work presents three main contributions. First, we show how an encoder model's limited context window can be mitigated through relevant text

Proceedings of the Fifteenth Language Resources and Evaluation Conference (LREC 2026), pages 8040–8055
11–16 May 2026.

selection. This allows encoder models to outperform any tested prompting method with decoder models. Text selection also reduces training and inference time for encoder and decoder models alike, with little to no performance loss.

Second, we demonstrate that fine-tuning sub-10B decoder models with PC objectives matches SFT and outperforms fine-tuned encoder models.

Third, we show that applying PC to SA outperforms previously tested approaches on additional sentiment classification benchmarks in Norwegian and English.

2. Related Work

We here present related work on ELSA, a challenging task for text classification. We present previous work on the limitations of naïve text chunking and of selecting relevant text segments for classification with encoder models. We finally present related work on the limitations of large instruction-tuned decoder models for such tasks, and how smaller decoder models can be fine-tuned for text classification.

2.1. Entity-Level Sentiment Analysis

Ben-Ami et al. (2015) introduce and motivate the task of ELSA. We apply their task description of identifying the document-level sentiment for each entity mentioned in each text. This is a less researched task within sentiment analysis. As the dataset used by Ben-Ami et al. (2015) seems not to be publicly available, we study two available datasets with entity-specific sentiment annotations at the overall, document level of the text.

Bastan et al. (2020) released PerSenT, an entity-specific dataset based on English news reports.

Rønningstad et al. (2024) released the Norwegian ELSA dataset, annotated by trained annotators, and provide initial baselines for modeling the ELSA sentiment classification.

2.2. Efficient text selection

One aspect of our research is how to enable classification of texts longer than 512 tokens by encoder models. Work by Sheng et al. (2025) and Cai et al. (2024) demonstrates the problems arising from chunking text into fixed-length segments that may be semantically broken. Park et al. (2022) motivate research on text classification with transformers for texts longer than the 512 tokens context window in general. They applied TextRank similarity search to select relevant sentences, but found this method to perform worse than selecting sentences at random.

Concurrently with our work, the approach of anchoring text selection to relevant entity mentions has been reported as successful by Rønningstad and Negi (2025) for classification of entity framing, and by Douglas et al. (2025) for ICD Code Prediction. The approach of training a model to classify sentences or paragraphs as relevant for entity sentiment is comparable to classifying datasets annotated for text segments that are salient for extractive summarization (Dernoncourt et al., 2018).

2.3. Decoder Models for Text Classification

A common method for text classification, including SA, using generative decoder models is through inference on an instruction-tuned model where the prompt contains a task description, the text, and the available category labels.

The shortcomings, both in terms of resource efficiency and performance scores, of prompting generative models for NLP classification tasks have been studied by Bucher and Martini (2024) in an English-only study comparing fine-tuned encoder models to zero-shot inference with recent large decoder models. Similar findings have been reported across various contexts (Kocoń et al., 2023; Wei et al., 2023; Saattrup Nielsen et al., 2025; Tang et al., 2023).

The results show in general that when adequate labeled training data is present, the much smaller encoder models tend to outperform even the largest decoder models on NLP classification tasks such as SA. At the time of writing, the encoder model NorBERT3-large is ranked highest among the 262 models of any size at the EuroEval NLP leaderboard that have been fully tested for Norwegian NLU tasks.[1] The DeBERTa-large (He et al., 2021) is likewise ranked highest among all 327 models ranked on the English NLU leaderboard, including up to 500B parameters instruction-tuned decoder models.

2.4. Fine-Tuning Decoder Models

The Supervised Fine-Tuning (SFT) training regime commonly used as part of instruction-tuning, can also be used for fine-tuning for classification tasks. However, it is not trivial to achieve better results through SFT with decoder models than what can be achieved with smaller and faster encoder models (He et al., 2025; Rønningstad and Negi, 2025).

Another part of the toolbox for instruction-tuning a language model is the reward model trained through pairwise comparison (Leike et al., 2018; Ouyang et al., 2022b). The method has been applied to other tasks where ranking or comparison between two alternatives is needed (Guzmán et al.,

[1] https://euroeval.com/leaderboards/Monolingual/norwegian/#__tabbed_1_4; accessed July 2025.

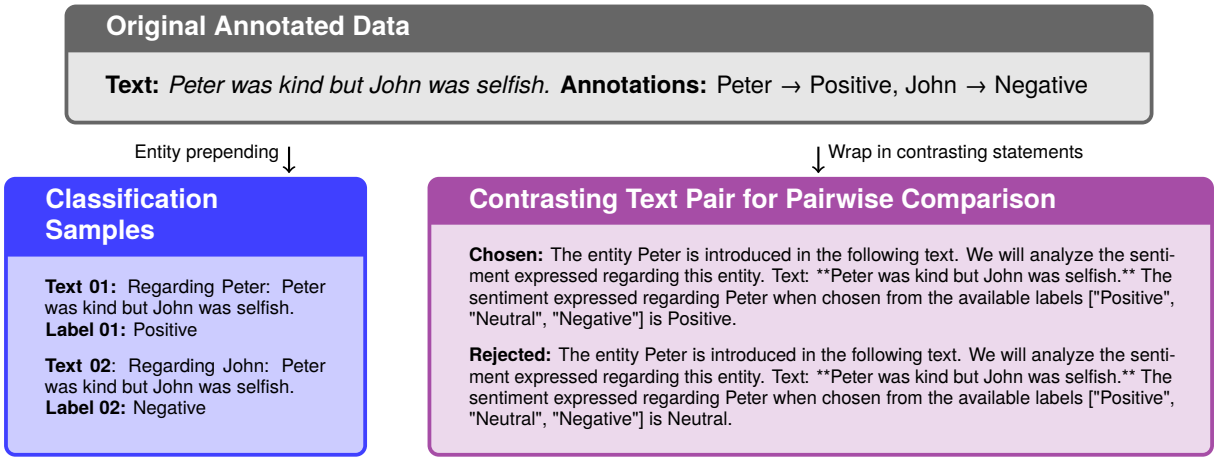

Figure 1: Data transformation from annotated text to training formats for the ELSA dataset. **Left**: Each entity is prepended to create a separate training sample for classification with encoders. **Right:** For PC, each entity initiates a text pair where the chosen text contains a correct sentiment judgment.

	PerSenT	ELSA
mean length	438.4	635.6
pct >512	25.2	59.4
# samples	3355	1908
# >512	845	1134

Table 1: Employed datasets for entity-specific sentiment classification with text spans exceeding 512 tokens. Values are for the training split.

Split	Negative	Neutral	Positive	Count
dev	41	145	138	324
test	21	132	94	247
train	241	1014	653	1908
Pct	12.22	52.08	35.70	

Table 2: Distribution of the 2479 sentiment-annotated entities in the ELSA dataset across splits and polarities.

2015; Moosa et al., 2024). We are not aware of previous work employing this method for SA, or for any text classification task in general.

3. Datasets

Our primary work is conducted on the Norwegian ELSA dataset. Our best performing methods are further tested on the English PerSenT dataset and on the Norwegian and English SA datasets incorporated in the EuroEval benchmark suite.

3.1. The ELSA Dataset

The ELSA dataset (Rønningstad et al., 2024) contains a subset of the Norwegian Review Corpus (NoReC, Velldal et al., 2018), where each text contains a review written by professional journalists. The ELSA dataset adds extensive entity-specific annotations, including sentiment labels for each entity (person or organization) on both the sentence- and document-level, annotated by trained human annotators. Critically for our work, as can be seen in Table 1, 60% of texts in the training split exceed 512 tokens. The ELSA dataset has the label distribution per entity as seen in Table 2. The initial baselines for modeling as reported by Rønningstad et al. are a weighted average F_1 of 68.1% with the NorBERT3-large encoder model by aggregating sentence-level entity-specific predictions, versus 73.3% with zero-shot prompting of GPT-4.

3.2. The PerSenT Dataset

The PerSenT dataset contains news articles selected from MPQA (Deng and Wiebe, 2015; Wiebe et al., 2005), the KBP Challenge (Ellis et al., 2014), and MediaRank (Ye and Skiena, 2019). In contrast to the Norwegian ELSA dataset, texts are annotated through crowdsourcing for sentiment regarding only one central entity. The sentiment regarding this entity is labeled at the document level, and for each of the first 16 paragraphs. 25% of the texts exceed 512 tokens.

The best-performing sentiment classifier reported in the original paper is a BERT classifier trained on the entire document, yielding a macro-averaged F_1 score of 48.38%. Their text selection methods yielded slightly weaker results than training on the entire document. As the dataset has two test splits, we refer only to evaluations on the Standard test split. Kuila and Sarkar (2024a) report on classification through prompting two sub-10B models, Mistral-7b and Falcon-7b-instruct. None of the zero-shot experiments performed better than the

BERT baseline, while their best few-shot approach yielded a macro F_1 of 50.45 with Falcon-7b-instruct.

3.3. The EuroEval Datasets

In order to further evaluate our method of pairwise comparison for sentiment classification, we select the Norwegian and English datasets for sentiment analysis (SA) in the EuroEval benchmarking framework. These contain sentences labeled as either Positive, Neutral or Negative. There is one dataset for Norwegian SA in the test suite, a subset of NoReC (Velldal et al., 2018). The sentence-level version was introduced by Kutuzov et al. (2021), based on the annotations by Øvrelid et al. (2020). The English SA dataset in the test suite is a subset of SST-5 (Socher et al., 2013). For both datasets, 1024 sentences were sampled from their training split, 256 sentences were sampled from the validation split, and 2048 sentences were sampled from the test split. The best performing sub-10B model for these tasks when accessed in July 2025 was the gemma-2-9b-it model with results as shown in Table 6.

4. Experiments

In the following, we present how relevant segments from the text are extracted in order to allow encoder models with a limited context window to classify longer texts, and also to make training processes less resource-demanding. The resulting texts, selected through heuristics and through modeling, are subsequently used for training and evaluating entity-level sentiment classifiers, and we compare with results obtained using the original fulltext version. Subsequently, the chosen pretrained language models are presented, with our methods for fine-tuning these for SA: categorical classification with encoder models, mere prompting of decoder models, and finally SFT and PC using decoder models. Our computational resources for these experiments are limited to one GPU with 64 GB of memory, and we did not consider models with 10 billion parameters or more. Code examples are available online.[2]

4.1. Text Selection Heuristics

The ELSA dataset is annotated for sentiment regarding each person and organization, which are standard entity categories in named entity recognition (NER). Suitable NER models exist for various languages, including Norwegian and English. We can therefore label any text containing these entities as part of the pre-processing steps and experiment with alternative heuristics for text selection based on sentences in which an entity is mentioned. The presence of named entities allows us to compare the following text selection heuristics: **ent2end**, which includes all text from the first mention of an entity to the end of the document; **ent2ent**, which spans from the sentence with a mention of the entity in question until a sentence occurs with a different entity mentioned. Lastly **entsent** selects only sentences where the entity is mentioned. Among these three, the ent2ent approach yielded best results from fine-tuning encoder models. Results from all three approaches are shown in Figure 2, while only the ent2ent results are reported in Table 3, which presents our results from all text selection methods and all modeling methods applied to the ELSA dataset. The formal procedure for extracting text spans based on the ent2ent heuristic is detailed in Algorithm 1 in Appendix B.

4.2. Predicting Relevant Sentences

The sentence-level annotations of the ELSA dataset supply ground truth labels identifying sentiment-relevant text with respect to the entity in question. As this information is not available for new texts, it is necessary to train a model on the annotated data in order to classify new sentences as relevant or irrelevant for sentiment.

To validate the potential of this method, we first train an entity-level sentiment classifier on a dataset where the sentences annotated for sentiment relevance with respect to the entity are concatenated as the training text. We found that training an entity sentiment predictor from the *annotated* relevant sentences yields strong results, achieving over 82% F_1, which is considerably better than previous work, as shown in Table 3. We therefore train a sentence relevance classifier based on the annotated relevance for each sentence with respect to each entity.

Formally, we have as input the ELSA dataset where sentence sentiment is annotated with respect to each entity. These are the relevant sentences that the selector model should learn to distinguish from the non-relevant sentences.

Input: Document collection $\mathcal{D}$, where each $d \in \mathcal{D}$ contains sentences $\{s_1, s_2, \ldots, s_m\}$ and entities $\{e_1, e_2, \ldots, e_k\}$.

Task: For each pair (s_i, e_j) where $s_i \in d$ and $e_j \in d$, predict: relevance$(s_i, e_j) \in \{\text{relevant}, \text{irrelevant}\}$

Through modeling sentence relevance, we create two new dataset versions for entity-specific SA: **relevant sentences:** Concatenate the predicted relevant sentences per entity; and **relevant span:** Concatenate all sentences in the text from the first to the last predicted relevant sentence. This latter

[2] https://github.com/egilron/elsa-lrec

Model name	Max l	Method	fulltext	ent2ent	rel sents	rel span	Mean
norbert3-large†	512	cc	68.1†				
GPT-4†		0-shot	73.3†				
norbert3-large	512	cc	67.23 (2.7)	72.20 (1.5)	77.04 (0.5)	73.15 (4.2)	72.41 (2.2)
norbert4-large	512	cc	72.77 (2.2)	77.22 (2.1)	76.07 (3.0)	77.91 (1.6)	75.99 (2.2)
norbert4-large	4096	cc	78.69 (1.2)	74.56 (1.7)	74.95 (2.1)	76.64 (3.4)	76.21 (2.1)
gemma-2-9b-it	8192	3-sh maj3	62.58	62.24	58.55	65.52	62.22
gemma-2-9b-it	8192	0-sh nosamp	73.14	67.24	63.79	64.55	67.18
gemma-2-9b-it	8192	sft	82.37 (1.3)	83.50 (1.0)	79.65 (3.7)	76.52 (1.6)	80.51 (1.9)
gemma-2-2b	4096	pc	78.18 (0.6)	78.84 (1.5)	77.73 (2.4)	76.76 (0.2)	77.88 (1.2)
gemma-2-9b	4096	pc	83.08 (0.9)	83.30 (0.3)	82.51 (1.5)	82.82 (0.3)	82.93 (0.8)

Table 3: Performance across models, classification methods and text selection method on the ELSA dataset. Mean weighted F_1 (st. dev.) over three runs. Max l: max input length; Method: cc: categorical classification, 3-sh maj3: 3-shot prompt, majority vote 3 runs, 0-sh nosamp: Zero-shot prompt, no sampling, pc: pairwise comparison, sft: supervised fine-tuning. (†) Previous work (Rønningstad et al., 2024). Table 4 shows training time.

	nb4	gemma-2-9b	
Text version	CC	PC	SFT
fulltext	23	703	1607
ent2ent	10	247	686

Table 4: Training duration in minutes for finetuning norbert4-large, and gemma-2-9b with PC and SFT.

approach has the potential of better capturing the semantic coherence, while risking more noise from sentiment signals regarding other entities.

We trained models for selecting relevant sentences from gemma-2-9b and NorBERT3. As the F_1 score was higher using the NorBERT3 model, this model was used to predict sentence relevance on the test set. To assess whether text selection introduces errors independent of the classifier, we manually inspected a sample of extracted texts for both the ent2ent and relevant span methods; the results are presented in Appendix D.

4.3. ELSA Classification with Encoders

As we have created alternative texts using the text selection methods described above, we now proceed to the task of entity-wise sentiment classification based on these text versions. We first describe the encoder-based experiments before proceeding to the decoder-based experiments.

Data Transformation. As one text may contain different sentiment regarding the various entities mentioned, the model needs a signal as to which entity is under consideration. We solve this by creating one training example from each entity mentioned, prepending the text with "Regarding <entity>:". See Figure 1 for example.

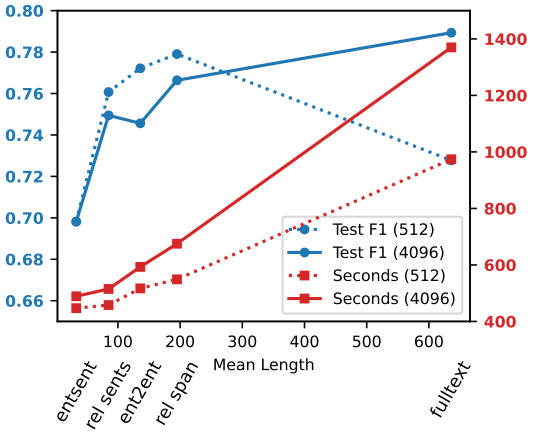

Figure 2: Weighted F_1 scores and training time for NorBERT4-large, max textlength set to 512 and 4096 tokens, on various ELSA dataset versions.

Categorical Classification. We here employ the standard approach for sentiment classification through the HuggingFace AutoModelForSequenceClassification,[3] which attaches a classification head to the model with one output node per label.

After initial testing of relevant models, we chose the Norwegian *NorBERT3-large* (323M params) encoder model (Samuel et al., 2023) and the subsequent *NorBERT4-large*[4] as the pretrained encoder models to fine-tune. The latter allows us to experiment with longer context windows through its sliding window attention (Beltagy et al., 2020), making experiments with a context window of 4096 possible, which is adequate for the full text lengths in the ELSA dataset.

[3] hf.co/docs/transformers/en/model_doc/auto
[4] hf.co/ltg/norbert4-large

4.4. Prompting Decoders

We proceed to the first ELSA experiments based on decoder models. The gemma-2 decoder models were selected based on their reported performance for Norwegian on the EuroEval benchmark, and from our comparisons between gemma-2-9b, Mistral-7B-v0.3, and the Norwegian normistral-7b-warm during initial experiments. gemma-2-9b, the slightly larger model performed better on these tests with the Norwegian ELSA dataset. We therefore kept gemma-2 for all subsequent experiments with decoder models.

We first performed zero-shot and three-shot inference with gemma-2-9b-it. For zero-shot inference we found the performance through greedy decoding to be better than majority voting over three inference runs using a low temperature setting, and we report this as our best result among the experiments of directly prompting the instruction-tuned gemma-2-9b-it model. Experiments with the non-instruction-tuned gemma-2-9b, and with the two gemma-2-2b versions yielded poor results and are not reported. The zero-shot prompt would be similar to the chosen example shown for PC in Figure 1, except for the last part where a label is not stated but asked for as reply. We also report our best results using a three-shot prompt, one example from each label. Here the best results were obtained from setting the temperature to 0.3 and selecting the majority vote label over three runs.

4.5. Supervised Fine-Tuning

SFT training uses the same next token prediction objective as pretraining, but the loss is only computed over the response part of a given prompt–response pair. In our SFT set-up, each training sample is a question–reply pair; the question is the same as what would be used for inference without fine-tuning and the reply is the true label.

To facilitate SFT of the gemma-2-9b-it model on a single 64 GB GPU instance, we used 4-bit quantization and low-rank adapters (LoRA). We conduct experiments varying the LoRA rank r between 16 and 256 and the adapter dimension a between 32 and 256, setting r to either half of or equal to a. We kept the batch size to 1, gradient accumulation steps to 4. We experimented with learning rates from 1×10^{-5} to 5×10^{-4}. Best results were achieved with a learning rate of 5×10^{-5} and LoRA settings $a = 256$ and $r = 128$, training for 22 epochs. The SFT fine-tuning setup appeared to be sensitive to hyperparameter settings, and many alternatives were tested before arriving at a well functioning setup. See Oliver and Wang (2024) and Pareja et al. (2024) for examples of related observations. Test results are from best epoch according to evaluation on the dev split.

4.6. Pairwise Comparison

When training a model for classification through pairwise comparison, the model is trained to score preferred texts higher than alternatives that are not preferred (rejected). A regression head is attached to the model, and two texts are passed through the model, one preferred text and one rejected text. Loss is calculated from comparing the outputs from the two texts, as will be described subsequently. As we have not seen previous work applying PC to text classification, we describe this method in some detail. We first describe how our dataset is transformed to fit the PC regime, then we present the loss function in more detail, before we describe our experimental setup for ELSA classification with gemma-2 and PC.

Data Transformation. For PC, we need two texts that are contrasted, one chosen and one rejected. The texts need not be conversational nor wrapped in a chat template. As shown in Figure 1, we wrap the entity and text in a statement describing the task, showing the possible categories. The text concludes with the true category for the chosen example, and with a wrong category for the rejected example.

As the classification task is to select between three categories, another pair may be created from the same training instance, using the other incorrect category in the rejected example. Initial experiments showed a small benefit from creating rejected texts using both incorrect categories, effectively doubling the training dataset size. During inference time, we pass one text per sentiment category through the model, and select the highest scoring category as the predicted label. See Appendix A for example implementation.

Loss Function. The loss function $\mathcal{L}(\theta)$ in our fine-tuning through PC is derived from the Bradley–Terry model (Bradley and Terry, 1952), which models the probability that one item is preferred over another as:

$$P(y_w \succ y_l|x) = \sigma\left(r_\theta(x, y_w) - r_\theta(x, y_l)\right) \quad (1)$$

where σ is the sigmoid function, $r_\theta(x, y)$ is the scalar output of the reward model for input x and response y. y_w is the winning (chosen) response, and y_l is the losing (rejected) response.

The loss function is formulated as the negative log-likelihood, implemented as the mean loss over batches of size B:

$$\mathcal{L}_B(\theta) = -\frac{1}{B}\sum_{i=1}^{B} \log \sigma(\Delta r_i) \quad (2)$$

where $\Delta r_i = r_\theta(x_i, y_{w,i}) - r_\theta(x_i, y_{l,i})$ represents the reward difference for the i-th training pair.

Model	Method	F_1
BERT†	fulltext	48.38†
Falcon-7b-instruct‡	Few-shot	50.45‡
gemma-2-9b	PC-ent2ent	52.80 (1.0)
gemma-2-9b	PC-fulltext	52.85 (0.6)

Table 5: Models for predicting entity-specific sentiment on the PerSenT dataset. Macro-averaged F_1. Our results: mean (st. dev.) over three runs. Our models are trained with the PC training objective, using the full text or the ent2ent version. (†) Previous work (Bastan et al., 2020), (‡) previous work (Kuila and Sarkar, 2024b).

This formulation is implemented in the Hugging Face TRL RewardTrainer (von Werra et al., 2022).

While our approach shares the pairwise comparison principle with *contrastive learning*, it differs in output structure: traditional contrastive learning learns multidimensional embeddings that are compared via distance metrics in the embedding space, whereas PC directly computes scalar preference scores.

Implementing Pairwise Comparison We fine-tuned the gemma-2-2b and gemma-2-9b models though pairwise comparison using text templates as demonstrated in Figure 1. Using non-instruction-tuned models allows for a wider selection of language-specific models, and it allows for a simpler prompt, avoiding the need for model-specific chat templates.

A selection of design choices were finalized for the PC method during initial experiments with the TRL RewardTrainer. We used Norwegian prompts for Norwegian text; created two chosen-rejected pairs per data item (one per incorrect label); trained for two epochs; and employed 4-bit quantization with LoRA (rank=16, alpha=32) to meet our GPU memory limitations. We did not experiment with alternative hyperparameters, as the default values resulted in successful fine-tuning. More implementation details are found in Appendix C.

4.7. Further Evaluations of the PC Method

To assess the broader applicability of the PC method for sentiment classification, we fine-tune gemma-2 models through PC on the PerSenT dataset, and on the Norwegian and English SA datasets from EuroEval.

PerSenT. We here performed the same ent2ent text selection as previously described, using a publicly available English NER pipeline[5] to find the mentions of the entity in question. We fine-tuned gemma-2-9b on both the ent2ent text and the full-text using PC. The experimental setup was the same as for the ELSA dataset, except for epochs which were set to 4.

Lang	Model	EuroEval	Ours
Norw	encoder	71.98 ±1.98	
Norw	g-2-2b	46.62 ±3.15	74.79 ±0.72
Norw	g-2-9b	75.82 ±0.92	**79.01 ±0.75**
Eng	encoder	62.94 ±3.0	
Eng	g-2-2b	65.91 ±0.99	**71.94 ±0.7**
Eng	g-2-9b	69.44 ±1.45	**72.92 ±0.87**

Table 6: Evaluation results for our fine-tuned gemma-2 models compared with the reported results on the EuroEval leaderboard, including a 95% confidence interval. Our models provide a new state-of-the-art for sub-10B parameter models on these datasets.

The EuroEval Benchmark. We replicate the EuroEval tests[6] for the task of sentiment analysis. See Table 6. The datasets are prepared for pairwise comparison training applying the EuroEval prompt templates.

We follow the test setup reported for the EuroEval test scores and bootstrap each dataset ten-fold. For each bootstrapped version, we train and test three models using three fixed seeds. The ten mean macro-averaged F_1 scores over three seeds are averaged, and a 95% confidence interval is calculated from these ten values.

5. Results and Discussion

Table 3 shows the results from the experiments on the ELSA dataset. The results for each modeling approach are further presented and discussed below.

5.1. ELSA with Encoder Models

For the encoder models with a 512-token context window, training on the (truncated when above 512 tokens) full text version, yields noticeably lower results than both the ent2ent text selection heuristics and the texts selected through relevant sentence prediction. However, the NorBERT4-large with a maximum text length of 4096 tokens is able to capture the entity-relevant sentiment marginally better from the full text version, than from any of the text

[5] hf.co/Gladiator/microsoft-deberta-v3-large_ner_conll2003

[6] euroeval.com/methodology

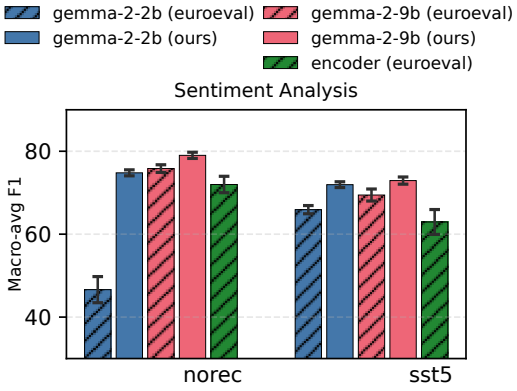

Figure 3: F_1 scores for EuroEval Norwegian and English sentiment analysis tasks. Striped bars show EuroEval baselines; solid bars show our fine-tuned models. Error bars indicate 95% confidence intervals based on 10 bootstrapped dataset versions.

selection methods tested. Figure 2 shows that this improvement comes at the cost of doubling training time, compared to the second best text selection method. Best weighted F_1 scores for NorBERT4-large are 78.69% on the full text version with the 4096 tokens context window, and 77.22% on the ent2ent version with the 512 tokens context window. This is a clear improvement over previous work, and over our own decoder inference through prompting. Table 4 reveals the remarkably shorter training times for our encoder experiments, and also how text selection cuts training times to less than half for all models. Figure 2 shows F_1 scores and training times for NorBERT4-large as a function of mean input text lengths from the various text selection methods. Text selection methods demonstrate their effectiveness by retaining sentiment signals within the 512-token constraint, while naïve truncation of text beyond the context window results in signal loss.

5.2. Prompting Decoders

The best performance from prompting the instruction-tuned model for the entity-specific sentiment label was obtained with a zero-shot prompt with no sampling, using the full document text as input. The scores were in general noticeably weaker than those of the fine-tuned models. We do not know why the results with 3-shot prompts were so low, but speculate that although the context window is large enough for a prompt with three examples of these longer texts, the complex sentiment signals are difficult to learn from, for a model this size. We do observe, however that with the relevant span text selection method, the three-shot experiment outperforms two of the zero-shot experiments. In general, these results support the observations reported in previous work that with training data available, fine-tuning an encoder model for text classification tends to outperform prompting an instruction-tuned decoder model.

5.3. Supervised Fine-Tuning

Table 3 shows that after extensive hyperparameter tuning (see Section 4.5), SFT achieved results surpassing both encoder-based fine-tuning and prompting instruction-tuned models. As shown in Table 4, SFT is the most compute-demanding method tested. Here as well, training time was reduced to less than half through text selection. Further, the ent2ent selection method yielded a not significant improvement over sending the full text.

5.4. Pairwise Comparison

We see how training gemma-2-9b for entity-level sentiment analysis with PC achieves the best results on average. Interestingly, the results are quite similar for all four methods reported in Table 3, reducing the importance of extracting relevant text before fine-tuning. Again, the ent2ent approach cuts training time to less than half of what is observed for the fulltext approach. We further notice that PC fine-tuning was not particularly dependent upon hyperparameter tuning, and the hyperparameters reported previously were largely our initial settings. In our experiments, PC finetuning on the ELSA dataset yielded more stable results than SFT, both across the three seeds, and across the four textual input alternatives. The mean test scores were slightly higher, less hyperparameter tuning was required, and training time was shorter here than with SFT.

5.5. PerSenT and EuroEval

The results in Table 5 show that fine-tuning gemma-2-9b with PC achieves a new state-of-the-art for the English PerSenT dataset, surpassing recent previous work with elaborate prompting techniques for sub-10B models.

The results in Table 6 and Figure 3 show that fine-tuning with PC allows the gemma-2 decoder models to excel at the sentence-level sentiment classification task as well, surpassing all the more than hundred sub-10B models tested on each of these datasets on the EuroEval leaderboard.

6. Conclusion

We have explored methods for modeling entity-level sentiment analysis where each text is up to 4096 tokens of length, using sub-10B language models. Our experiments confirm the strong performance of

encoder models, when the limitations of the context window is mitigated. We show that our proposed ent2ent method for entity-relevant text extraction reduces training- and inference time considerably on longer texts, with a minimal loss of performance on the ELSA task.

We further show that decoder models can obtain considerably better performance when fine-tuned for this task as well, and that the PC method is a robust alternative to SFT, obtaining similar to better results with shorter training and minimal hyperparameter tuning. These fine-tuned models provide a new state-of-the-art on the additional datasets tested. Our results suggest that the PC method may be relevant for other NLP text classification tasks as well.

7. Acknowledgements

The work documented in this publication has been carried out within the NorwAI Centre for Research-based Innovation, funded by the Research Council of Norway (RCN), with grant number 309834.

We acknowledge the EuroHPC Joint Undertaking for awarding this project access to the EuroHPC supercomputer LUMI, hosted by CSC (Finland) and the LUMI consortium through a EuroHPC Regular Access call.

Computations were performed in part on resources provided through Sigma2 – the national research infrastructure provider for High-Performance Computing and large-scale data storage in Norway.

We thank Lucas Georges Gabriel Charpentier and David Samuel for their contributions to our codebase and for their help with deploying our experiments on the LUMI supercomputer.

8. Bibliographical References

Jason Ansel, Edward Yang, Horace He, Natalia Gimelshein, Animesh Jain, Michael Voznesensky, Bin Bao, Peter Bell, David Berard, Evgeni Burovski, et al. 2024. Pytorch 2: Faster machine learning through dynamic python bytecode transformation and graph compilation. In Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2, pages 929–947.

Jeremy Barnes, Laura Oberlaender, Enrica Troiano, Andrey Kutuzov, Jan Buchmann, Rodrigo Agerri, Lilja Øvrelid, and Erik Velldal. 2022. SemEval 2022 task 10: Structured sentiment analysis. In Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022), pages 1280–1295, Seattle, United States. Association for Computational Linguistics.

Mohaddeseh Bastan, Mahnaz Koupaee, Youngseo Son, Richard Sicoli, and Niranjan Balasubramanian. 2020. Author's sentiment prediction. In Proceedings of the 28th International Conference on Computational Linguistics, pages 604–615, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. arXiv preprint arXiv:2004.05150.

Zvi Ben-Ami, Ronen Feldman, and Benjamin Rosenfeld. 2015. Exploiting the focus of the document for enhanced entities' sentiment relevance detection. In 2015 IEEE International Conference on Data Mining Workshop (ICDMW), pages 1284–1293.

Ralph Allan Bradley and Milton E Terry. 1952. Rank analysis of incomplete block designs: I. the method of paired comparisons. Biometrika, 39(3/4):324–345.

Martin Juan José Bucher and Marco Martini. 2024. Fine-tuned'small'llms (still) significantly outperform zero-shot generative ai models in text classification. arXiv preprint arXiv:2406.08660.

Hongjie Cai, Heqing Ma, Jianfei Yu, and Rui Xia. 2024. A joint coreference-aware approach to document-level target sentiment analysis. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 12149–12160, Bangkok, Thailand. Association for Computational Linguistics.

Lingjia Deng and Janyce Wiebe. 2015. MPQA 3.0: An entity/event-level sentiment corpus. In Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 1323–1328, Denver, Colorado. Association for Computational Linguistics.

Franck Dernoncourt, Mohammad Ghassemi, and Walter Chang. 2018. A repository of corpora for summarization. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), Miyazaki, Japan. European Language Resources Association (ELRA).

James C. Douglas, Yidong Gan, Ben Hachey, and Jonathan K. Kummerfeld. 2025. Less is

more: Explainable and efficient ICD code prediction with clinical entities. In Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 30835–30847, Vienna, Austria. Association for Computational Linguistics.

Joe Ellis, Jeremy Getman, and Stephanie M Strassel. 2014. Overview of linguistic resources for the tac kbp 2014 evaluations: Planning, execution, and results. In Proceedings of TAC KBP 2014 Workshop, National Institute of Standards and Technology, pages 17–18.

Francisco Guzmán, Shafiq Joty, Lluís Màrquez, and Preslav Nakov. 2015. Pairwise neural machine translation evaluation. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 805–814, Beijing, China. Association for Computational Linguistics.

Charles R Harris, K Jarrod Millman, Stéfan J Van Der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J Smith, et al. 2020. Array programming with numpy. nature, 585(7825):357–362.

Mingqian He, Fei Zhao, Chonggang Lu, Ziyan Liu, Yue Wang, and Haofu Qian. 2025. Gencls++: Pushing the boundaries of generative classification in llms through comprehensive sft and rl studies across diverse datasets. arXiv preprint arXiv:2504.19898.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, pages 168–177.

J. D. Hunter. 2007. Matplotlib: A 2d graphics environment. Computing in Science & Engineering, 9(3):90–95.

Jan Kocoń, Igor Cichecki, Oliwier Kaszyca, Mateusz Kochanek, Dominika Szydło, Joanna Baran, Julita Bielaniewicz, Marcin Gruza, Arkadiusz Janz, Kamil Kanclerz, et al. 2023. Chatgpt: Jack of all trades, master of none. Information Fusion, 99:101861.

Alapan Kuila and Sudeshna Sarkar. 2024a. Deciphering political entity sentiment in news with large language models: Zero-shot and few-shot strategies. In Proceedings of the Second Workshop on Natural Language Processing for Political Sciences @ LREC-COLING 2024, pages 1–11, Torino, Italia. ELRA and ICCL.

Alapan Kuila and Sudeshna Sarkar. 2024b. Deciphering political entity sentiment in news with large language models: Zero-shot and few-shot strategies. arXiv preprint arXiv:2404.04361.

Andrey Kutuzov, Jeremy Barnes, Erik Velldal, Lilja Øvrelid, and Stephan Oepen. 2021. Large-scale contextualised language modelling for norwegian. In Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa 2021).

Jan Leike, David Krueger, Tom Everitt, Miljan Martic, Vishal Maini, and Shane Legg. 2018. Scalable agent alignment via reward modeling: a research direction. arXiv preprint arXiv:1811.07871.

Yun Luo, Hongjie Cai, Linyi Yang, Yanxia Qin, Rui Xia, and Yue Zhang. 2022. Challenges for open-domain targeted sentiment analysis.

Ibraheem Muhammad Moosa, Rui Zhang, and Wenpeng Yin. 2024. Mt-ranker: Reference-free machine translation evaluation by inter-system ranking. arXiv preprint arXiv:2401.17099.

Michael Oliver and Guan Wang. 2024. Crafting efficient fine-tuning strategies for large language models. arXiv preprint arXiv:2407.13906.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022a. Training language models to follow instructions with human feedback. In Advances in Neural Information Processing Systems, volume 35, pages 27730–27744. Curran Associates, Inc.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022b. Training language models to follow instructions with human feedback. Advances in neural information processing systems, 35:27730–27744.

Lilja Øvrelid, Petter Mæhlum, Jeremy Barnes, and Erik Velldal. 2020. A fine-grained sentiment dataset for Norwegian. In Proceedings of the 12th Edition of the Language Resources and Evaluation Conference, Marseille, France, 2020.

The pandas development team. 2020. pandas-dev/pandas: Pandas.

Aldo Pareja, Nikhil Shivakumar Nayak, Hao Wang, Krishnateja Killamsetty, Shivchander Sudalairaj, Wenlong Zhao, Seungwook Han, Abhishek Bhandwaldar, Guangxuan Xu, Kai Xu, et al. 2024. Unveiling the secret recipe: A guide for supervised fine-tuning small llms. arXiv preprint arXiv:2412.13337.

Hyunji Park, Yogarshi Vyas, and Kashif Shah. 2022. Efficient classification of long documents using transformers. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 702–709, Dublin, Ireland. Association for Computational Linguistics.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12:2825–2830.

Qiankun Pi, Jicang Lu, Taojie Zhu, Yepeng Sun, Shunhang Li, and Jiaxing Guo. 2024. Enhancing cross-evidence reasoning graph for document-level relation extraction. PeerJ Computer Science, 10:e2123.

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. SemEval-2014 task 4: Aspect based sentiment analysis. In Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), pages 27–35, Dublin, Ireland. Association for Computational Linguistics.

Egil Rønningstad, Roman Klinger, Lilja Øvrelid, and Erik Velldal. 2024. Entity-level sentiment: More than the sum of its parts. In Proceedings of the 14th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis, pages 84–96, Bangkok, Thailand. Association for Computational Linguistics.

Egil Rønningstad and Gaurav Negi. 2025. Ltg at semeval-2025 task 10: Optimizing context for classification of narrative roles.

Dan Saattrup Nielsen, Kenneth Enevoldsen, and Peter Schneider-Kamp. 2025. Encoder vs decoder: Comparative analysis of encoder and decoder language models on multilingual NLU tasks. In Proceedings of the Joint 25th Nordic Conference on Computational Linguistics and 11th Baltic Conference on Human Language Technologies (NoDaLiDa/Baltic-HLT 2025), pages 561–572, Tallinn, Estonia. University of Tartu Library.

David Samuel, Andrey Kutuzov, Samia Touileb, Erik Velldal, Lilja Øvrelid, Egil Rønningstad, Elina Sigdel, and Anna Palatkina. 2023. NorBench – a benchmark for Norwegian language models. In Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 618–633, Tórshavn, Faroe Islands. University of Tartu Library.

Boheng Sheng, Jiacheng Yao, Meicong Zhang, and Guoxiu He. 2025. Dynamic chunking and selection for reading comprehension of ultra-long context in large language models. In Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 31857–31876, Vienna, Austria. Association for Computational Linguistics.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Yixuan Tang, Yi Yang, Allen Huang, Andy Tam, and Justin Tang. 2023. FinEntity: Entity-level sentiment classification for financial texts. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pages 15465–15471, Singapore. Association for Computational Linguistics.

Erik Velldal, Lilja Øvrelid, Eivind Alexander Bergem, Cathrine Stadsnes, Samia Touileb, and Fredrik Jørgensen. 2018. NoReC: The Norwegian review corpus. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), Miyazaki, Japan. European Language Resources Association (ELRA).

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, and Nathan Lambert. 2022. Trl: Transformer reinforcement learning. https://github.com/huggingface/trl.

Binghai Wang, Rui Zheng, Lu Chen, Zhiheng Xi, Wei Shen, Yuhao Zhou, Dong Yan, Tao Gui, Qi Zhang, and Xuanjing Huang. 2024. Reward modeling requires automatic adjustment based on data quality. In Findings of the Association for Computational Linguistics: EMNLP 2024, pages

4041–4064, Miami, Florida, USA. Association for Computational Linguistics.

Xiang Wei, Xingyu Cui, Ning Cheng, Xiaobin Wang, Xin Zhang, Shen Huang, Pengjun Xie, Jinan Xu, Yufeng Chen, Meishan Zhang, et al. 2023. Chatie: Zero-shot information extraction via chatting with chatgpt. arXiv preprint arXiv:2302.10205.

Wes McKinney. 2010. Data Structures for Statistical Computing in Python. In Proceedings of the 9th Python in Science Conference, pages 56 – 61.

Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. Language resources and evaluation, 39(2):165–210.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. arXiv preprint arXiv:1910.03771.

Junting Ye and Steven Skiena. 2019. Mediarank: Computational ranking of online news sources. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pages 2469–2477.

9. Language Resource References

Bastan, Mohaddeseh and Koupaee, Mahnaz and Son, Youngseo and Sicoli, Richard and Balasubramanian, Niranjan. 2020. PerSenT. Stony Brook University. https://stonybrooknlp.github.io/PerSenT/.

Lilja Øvrelid and Petter Mæhlum and Jeremy Barnes and Erik Velldal. 2020. NoReC sentence. Language Technology Group, University of Oslo. https://huggingface.co/datasets/ltg/norec_sentence.

Rønningstad, Egil and Øvrelid, Lilja and Velldal, Erik. 2024. ELSA: An Entity-Level Sentiment Analysis Dataset for Norwegian. Language Technology Group, University of Oslo. https://github.com/ltgoslo/ELSA.

Richard Socher and Alex Perelygin and Jean Wu and Jason Chuang and Christopher Manning and Andrew Ng and Christopher Potts. 2013. Stanford Sentiment Treebank. The Stanford Natural Language Processing Group. https://nlp.stanford.edu/sentiment/.

Appendix A. Pairwise Comparison Training for ELSA Sentiment classification

When implementing Pairwise Comparison for training a classifier to categorize the sentiment expressed regarding a given entity, we wrap the entity, the text and the label into one prompt, to be compared with the same entity and text, but with another label. The English version of our prompt template for creating sentiment classification pairs is as follows:

```
'The entity {0} is introduced in the Norwegian Main text. We will analyze the
    sentiment expressed regarding this entity.\nMain text: {1}\nThe sentiment
    expressed regarding {0} when chosen from the available labels `["Positive", "
    Neutral", "Negative"]` is {2}'
Entity mention: {0}
Main text: {1}
Label: {2}
```

Each text with its sentiment label is made into two training pairs. The only difference lies in the final label category word. English machine translation of the Norwegian text.

Chosen text 1

```
The entity Norah Jones is introduced in the Norwegian Main text.
We will analyze the sentiment expressed regarding this entity.
Main text: **But Norah Jones does backing vocals.
That's at least something.**
The sentiment expressed regarding Norah Jones when chosen from the
available labels ["Positive", "Neutral", "Negative"] is Positive
```

Rejected text 1

```
The entity Norah Jones is introduced in the Norwegian Main text.
We will analyze the sentiment expressed regarding this entity.
Main text: **But Norah Jones does backing vocals.
That's at least something.**
The sentiment expressed regarding Norah Jones when chosen from the
available labels ["Positive", "Neutral", "Negative"] is Negative
```

Chosen text 2

```
The entity Norah Jones is introduced in the Norwegian Main text.
We will analyze the sentiment expressed regarding this entity.
Main text: **But Norah Jones does backing vocals.
That's at least something.**
The sentiment expressed regarding Norah Jones when chosen from the
available labels ["Positive", "Neutral", "Negative"] is Positive
```

Rejected text 2

```
The entity Norah Jones is introduced in the Norwegian Main text.
We will analyze the sentiment expressed regarding this entity.
Main text: **But Norah Jones does backing vocals.
That's at least something.**
The sentiment expressed regarding Norah Jones when chosen from the
available labels ["Positive", "Neutral", "Negative"] is Neutral
```

Appendix B. Algorithms

Algorithm 1 describes the relevant text selection heuristics used in the *ent2ent* method. Algorithm 2 describes how a text is prepared for sentence selection model training through pairwise comparison between chosen and rejected text.

Algorithm 1 Extracting ent2ent text spans for classification

ent2enttexts $\leftarrow \emptyset$
for each document d in dataset D **do**
  `all_entitymentions` $\leftarrow$ indices for all entity-mentioning sentences in d
  for each entity e mentioned in d **do**
    selectedsentences $\leftarrow \emptyset$
    `entitymentions` $\leftarrow$ indices for sentences in d mentioning this entity
    for each sentece index i in `entitymentions` **do**
      selectedsentences $\leftarrow$ *selectedsentences* $\cup \{i\}$
      $i \leftarrow i + 1$
      while $i \notin$ `all_entitymentions` **do**
        selectedsentences $\leftarrow$ *selectedsentences* $\cup \{i\}$
        $i \leftarrow i + 1$
      end while
    end for
    ent2enttexts $\leftarrow$ *ent2enttexts* $\cup \{(e,$ *selectedsentences*$)\}$
  end for
end for

Algorithm 2 Generating Pairwise Dataset for selection model training

Function: `CreatePrompt(context, new_sentence, relevance)`
  Returns formatted prompt by concatenating inputs within template

for each document d in dataset D **do**
  for each entity e mentioned in d **do**
    `pairwise_dataset` $\leftarrow \emptyset$
    `context` $\leftarrow$ first sentence in d that contains a mention of e
    for each subsequent sentence `new_sentence` **do**
      Assign the annotated relevance label to variable `relevant`
      `chosen_example` $\leftarrow$ `CreatePrompt(context, new_sentence, relevant)`
      `rejected_example` $\leftarrow$ `CreatePrompt(context, new_sentence,` $\neg$`relevant)`
      `pairwise_dataset` $\leftarrow$ `pairwise_dataset` $\cup \{$`(chosen_example, rejected_example)`$\}$
      if `new_sentence` is labeled as relevant **then**
        `context` $\leftarrow$ `context` $+$ `new_sentence`
      end if
    end for
  end for
end for

Appendix C. Implementation Details

Our experiments were implemented in Python 3.11. All package versions are available on line,[7] where programming code will be added as well. We made extensive use of the following packages: transformers (Wolf et al., 2019), trl (von Werra et al., 2022), torch (Ansel et al., 2024), numpy (Harris et al., 2020) pandas (Wes McKinney, 2010; pandas development team, 2020), scikit-learn (Pedregosa et al., 2011) and matplotlib (Hunter, 2007).

Implementation Details for the Pairwise Comparison Training

Norwegian or English prompts A prompt in this context is the standard text repeated in Section Appendix A, which is attached to each text segment and entity. Table 9 shows a slight advantage with keeping the prompt in Norwegian, and this choice is kept for subsequent classification experiments. This is in line with the EuroEval evaluation setup, where prompts are kept in the language of the dataset evaluated on.

One or two negative examples While the selection model has only two labels to distinguish, *Relevant* or *Irrelevant*, for sentiment classification there are three labels to distinguish, *Positive*, *Neutral* or *Negative*. While the chosen text must contain the correct label, there are two options for the negative label. We tested two approaches for this: a) As rejected text, sample from the two incorrect labels according to these labels' distribution in the train set. b) Create two training pairs per entity. Each with the correct label in chosen text, and for the rejected text use the two incorrect labels, one in each pair. This second approach doubles the training set and therefore training time. Table 7 shows that using both incorrect labels gives a slight performance improvement.

Epochs of training While one epoch of training is considered enough for training reward models for RLHF to avoid overfitting (Ouyang et al., 2022a; Wang et al., 2024), we check for benefits from training longer, as our models need not respond to the same textual diversity as reward models for RLHF. Table 8 shows that two epochs of training yielded the best evaluation accuracy and the lowest evaluation loss.

Quantization and PEFT: For parameter-efficient fine-tuning (PEFT), we employed Low-Rank Adaptation (LoRA) with a rank parameter of 16 and an alpha scaling factor of 32. The values were chosen according to common practice and initial experiments with higher values that did not yield any improvements. We tested the impact of quantization. The 16-bit precision experiments yielded on average 82.87% accuracy, while the 4-bit quantized counterparts yielded 82.25%.

[7] https://github.com/egilron/elsa-lrec

	Pair per entity 1	2
Neg F_1	56.82	58.11
Pos F_1	81.38	82.40
W Avg F_1	80.15	**81.72**
F_1 Std (3 runs)	0.47	0.59

Table 7: Mean F_1 scores in % for training on one pair of chosen and rejected text per entity versus two pairs. The chosen text labels the text with correct label while rejected text labels the text incorrectly. All prompts are Norwegian. Text selection is "Relevant span".

epoch	eval loss	eval accuracy
1	0.4337	0.8472
2	0.3988	**0.8750**
3	1.1496	0.8596
4	1.0391	0.8688

Table 8: Evaluation loss and accuracy for a reward-model trained on preferring the correct sentiment label over an incorrect label.

Text selection	**English**	**Norwegian**
Entire document	83.95%	81.79%
Relevant only	81.48%	83.33%
Relevant span	81.17%	85.19%
Average	82.20%	83.44%

Table 9: Accuracy for predicting ELSA sentiment using pairwise comparison setup with gemma-2-9b. Wrapping the text in a Norwegian prompt yielded best results on average, and we kept this approach for later experiments.

Appendix D. Compression error inspection

We evaluated the impact of text selection methods on sentiment signal preservation, independent of the classification step. To do so, we manually inspected a sample of 50 non-neutral entities from the test set. For each entity, we compared the full article text with the text extracted using two methods: *Relevant span* and *ent2ent* (Sections 4.1 and 4.2).

Sentiment signal	Rel span count	Rel span comp	Ent2ent count	Ent2ent comp
lost	0	–	2	2.5%
weakened	7	22.8%	5	14.8%
intact	43	64.0%	43	39.9%

Table 10: Sentiment signal preservation and mean length (as a percentage of the full text) for 50 non-neutral entities, comparing the *Relevant span* and *ent2ent* text extraction methods.

The extracted texts were classified as having the sentiment signal intact, weakened, or lost. The results are presented in Table 10. For 43 of the 50 entities, the sentiment signal remained intact, despite a reduction in text length to 40–64% of the original. In cases where the signal was weakened, some relevant sentiment expressions were omitted, but sufficient context remained to convey the non-neutral sentiment. In the two instances where the signal was lost, the ent2ent method removed all relevant sentiment expressions. Notably, these expressions appeared in sentences preceding the entity's first mention by name.

For example, in one such case, the extracted text (machine-translated into English) reads:

There are many heartwarming moments and endearing fathers speaking about how much their daughters mean to them. "Maybe she can even become proud of me", says Sune Hansen from Tromsø.

Here, the entity *Sune Hansen* is introduced only in the final sentence, while the positive sentiment is expressed earlier. Since the ent2ent heuristic begins text selection at the first sentence containing the entity, this positive signal was lost. Despite these limitations, the ent2ent method still facilitates improved modeling, as reported in Table 3. We conclude that, for the majority of texts where the sentiment signal is preserved, the reduction in text length (approximately 50%) enhances classification performance, outweighing the occasional degradation introduced by this step.