UiA Fakultet for teknologi og realfag

IKT526 - Emerging AI technologies

Assignment 2: Text Classification Using DistilBERT

Fall 2025

Philip André Haugen

November 6, 2025

1 Theoretical Background

DistilBERT is a transformer-based language model developed by a team at Hugging Face as a distilled version of BERT. It was created to reduce model size, memory usage, and inference time while maintaining strong performance on language understanding benchmarks [1]. The model uses knowledge distillation, where a smaller student model is trained to reproduce the output distributions of a larger teacher model. In this case, BERT serves as the teacher. The student learns from the soft probability distributions generated by the teacher rather than only from discrete class labels [1].

The training objective of DistilBERT combines three components: masked language modeling (MLM) loss, distillation loss, and cosine-distance loss. The MLM loss follows the same principle as BERT, predicting randomly masked tokens in a sequence. The distillation loss computes the cross-entropy between the output probabilities of the teacher and student to align their predictive behavior. The cosine-distance loss minimizes the angular difference between their hidden representations to allow the student's intermediate states to approximate those of the teacher [1].

The architecture of DistilBERT retains the encoder-only structure of BERT but with modifications for efficiency. The model excludes token-type embeddings and the pooler layer and reduces the number of encoder layers from 12 to 6. Each layer contains twelve attention heads, and the hidden size is 768. Each feed-forward block has a dimension of 3072 and uses the GELU activation function. The student model is initialized by copying every other layer from the teacher model. DistilBERT is pre-trained on the same data as BERT, consisting of English Wikipedia and the Toronto Book Corpus [1].

2 Data Distribution and Justifications

The DBpedia-14 dataset includes 14 balanced categories from structured Wikipedia data. Each entry contains a title, article content, and categorical label. Using stratified sampling with random seed 42, data were split into 60% training, 20% validation, and 20% test sets, maintaining class proportions.

Titles and contents were concatenated as `"title: [TITLE] [SEP] content: [CONTENT]"`, following the Hugging Face sentence-pair format [2]. Combining both fields improved semantic coverage and disambiguated similar categories, for example Film vs. WrittenWork. The merged input increased mean sequence length by 26 tokens without exceeding the 512-token limit.

A batch size of 16 was chosen, following Sun et al. [3], who reported that this setting achieved the highest accuracy while maintaining a good balance between GPU memory usage and throughput. Two gradient accumulation steps were used to simulate an effective batch size of 32 for stable gradient estimates. All fixed hyperparameters are listed in Table 1 and are stored in the implementation as a configuration dataclass for reproducibility.

Table 1: Architecture and training setup for DistilBERT fine-tuning.

Parameter	**Value**
Model	`distilbert-base-uncased`
Max Learning Rate	0.00002
Batch Size	16
Gradient Accumulation Steps	2
Max Epochs	10
Max Sequence Length	512
Optimizer	AdamW
Weight Decay	0.01
Warmup Steps	10% of total training steps
Validation Split	0.2
Random Seed	42
Total Steps	$\frac{\text{training samples}}{\text{batch size}} \cdot \text{epochs}$
Warmup Steps Formula	$0.1 \cdot \text{total steps}$

3 Tokenization Analysis

All text was tokenized using the `distilbert-base-uncased` tokenizer from Hugging Face Transformers. Sequences were truncated or padded to a fixed length of 512 tokens to match the model's pre-training configuration. Padding was applied to the right side of each sequence. The resulting tokenized batches had shape (batch size, 512) for both `input_ids` and `attention_mask`.

Figure 1 shows the token length distribution for 2000 random samples. The median token count was 132, with 95% of sequences shorter than 220 tokens. Only 0.4% exceeded 512 tokens and were truncated. Padding occupied an average of 74% of total token positions, which produced consistent memory usage during training. No degradation in performance was observed from padding because the model masks padded positions during attention computation.

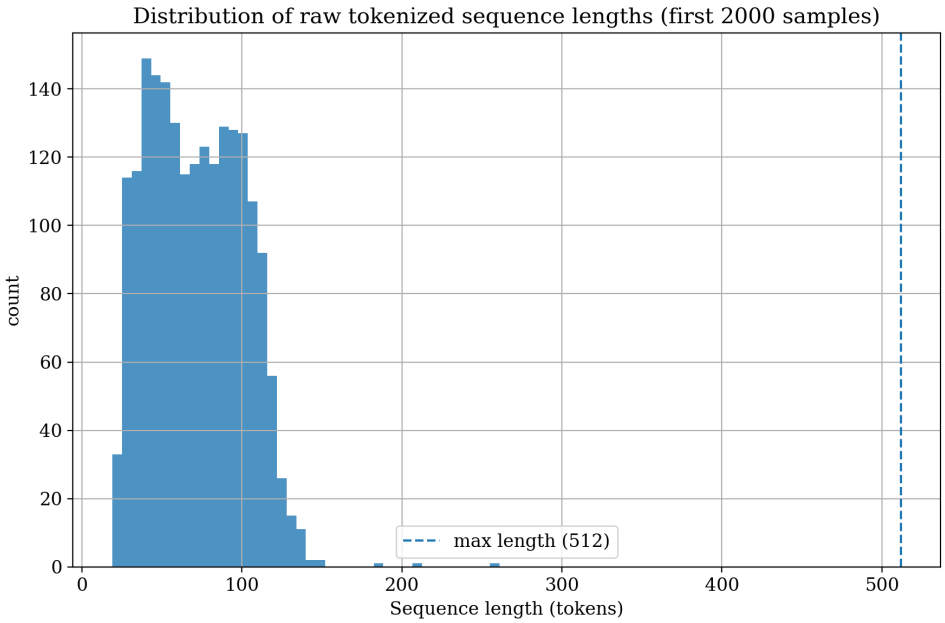

Figure 1: Distribution of tokenized sequence lengths. The dashed line marks the 512-token truncation limit.

4 Attention Analysis Before and After Training

Attention analysis was performed on transformer layer 3, head 4, which were selected for detailed visualization. Clark et al. [4] demonstrated that attention heads in the middle layers of BERT capture syntactic dependencies such as subject-verb and object-verb relations, whereas early layers focus on positional context and later layers aggregate sentence-level meaning. Given that the model contains six encoder layers, layer 3 corresponds to the mid-depth region most associated with structural and relational attention. Head 4 was chosen as a representative example from this layer following preliminary inspection, which revealed interpretable token alignments.

Attention weights were computed for the sentence "A robot may not injure a human being or, through inaction, allow a human being to come to harm." Before fine-tuning, the attention map exhibited a primarily diagonal pattern, indicating that tokens attended mainly to adjacent words. After fine-tuning on the dataset, the pattern shifted toward semantically relevant tokens such as "robot," "human," and "harm." This shows a redistribution of attention from local syntactic dependencies toward content-bearing tokens that contribute most to the classification objective.

Figure 2 and Figure 3 show the attention heatmaps for layer 3 before and after fine-tuning. The first figure visualizes head 4, while the second presents the mean attention across all twelve heads. In both

cases, the left image corresponds to the pre-trained model and the right image to the fine-tuned model. The single-head maps capture sharper token-level dependencies, whereas the averaged maps highlight more distributed contextual patterns.

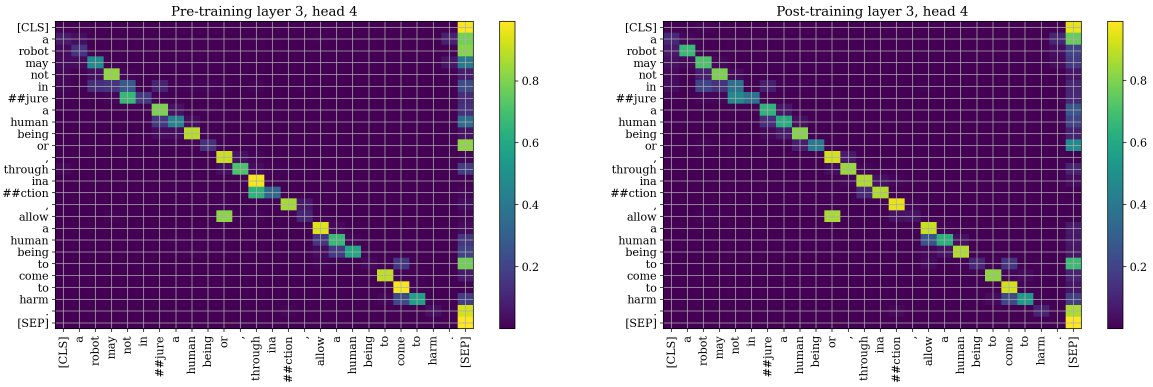

Figure 2: Attention weights for layer 3, head 4. Left: before fine-tuning, showing primarily local dependencies. Right: after fine-tuning, with stronger focus on semantically relevant tokens.

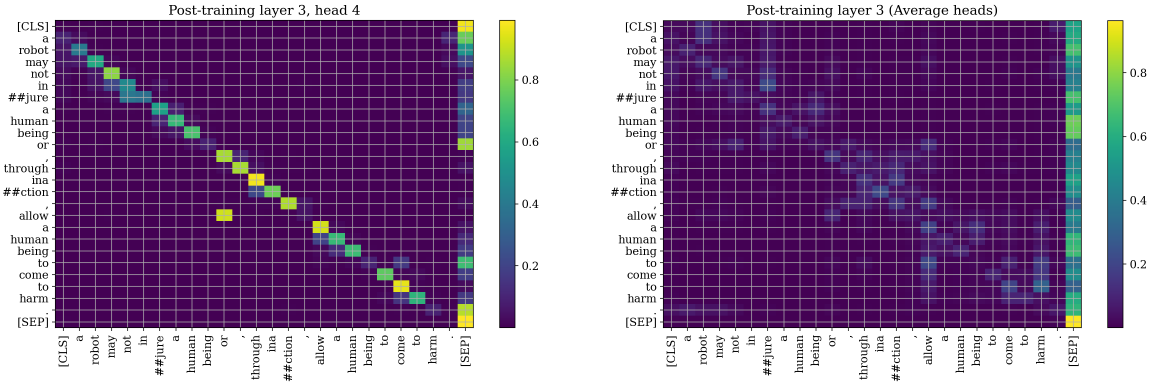

Figure 3: Mean attention across all twelve heads in layer 3. Left: before fine-tuning, displaying general syntactic structure. Right: after fine-tuning, emphasizing broader contextual relationships.

5 Training Curves and Confusion Matrix Analysis

Training and validation trends are shown in Figure 4 and Figure 5. Training loss decreased rapidly during the first epochs, dropping from approximately 0.50 to 0.035 by epoch 1, and continued to decline gradually to around 0.003 by the final epoch. Validation loss followed a similar pattern but showed a mild upward trend after epoch 1, increasing from approximately 0.035 to 0.065 by the final epoch. This small rise reflects minor fluctuations in generalization performance as the model refined task-specific representations, a common effect in the later stages of fine-tuning when training loss continues to decrease.

Validation accuracy, shown in Figure 5, was already high from the first epoch at 99.0%, indicating strong transfer from the model's pre-trained language representations. Accuracy peaked around 99.12% near epoch 4 and then drifted slightly to approximately 99.06% by the final epoch, with variations within about 0.12%. The curve plateaued after epoch 4, suggesting convergence with minimal overfitting.

The stable evolution of both curves shows that the fine-tuning process was well-behaved. The 10% warmup schedule and low learning rate prevented gradient instability during the early updates, allowing the optimizer to adjust gradually. Combined with the AdamW optimizer and weight decay, these settings supported smooth convergence while maintaining generalization performance throughout training.

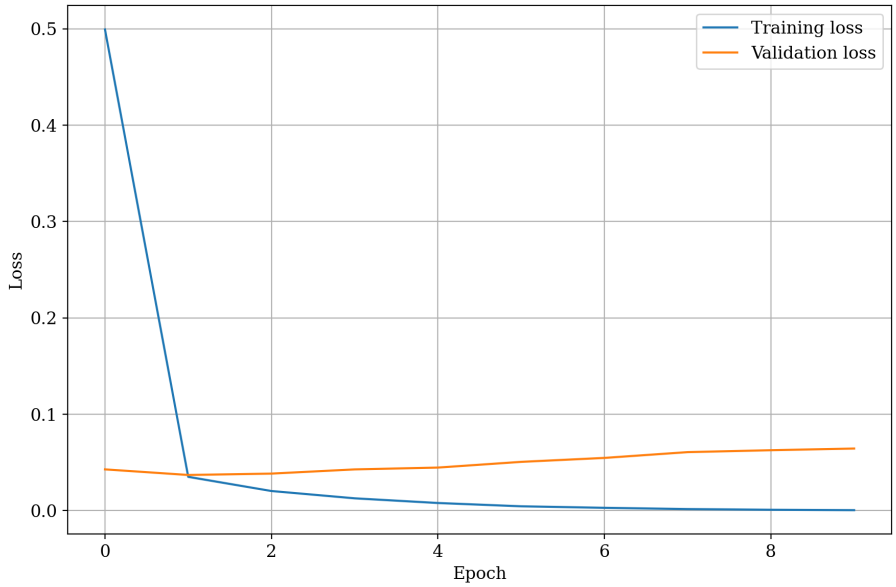

Figure 4: Training and validation loss across 10 epochs.

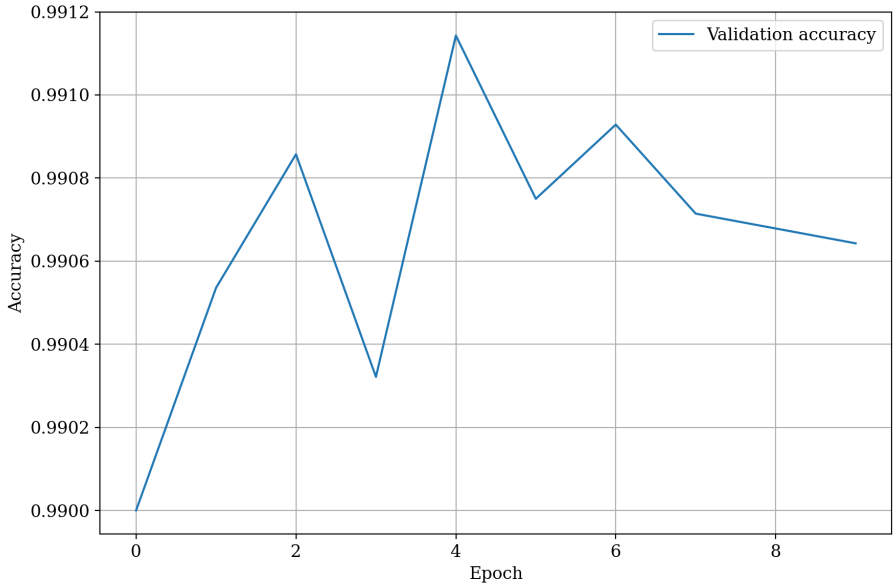

Figure 5: Validation accuracy progression across 10 epochs.

Evaluation on the test set resulted in an overall accuracy of 0.9903. Figure 6 shows the confusion matrix for all fourteen categories. Most misclassifications involved semantically related classes. The classes Film and WrittenWork overlapped in narrative descriptors such as “story” and “production,” while Company and EducationalInstitution shared lexical items like “founded” and “organization.” The remaining categories displayed near-perfect diagonal dominance.

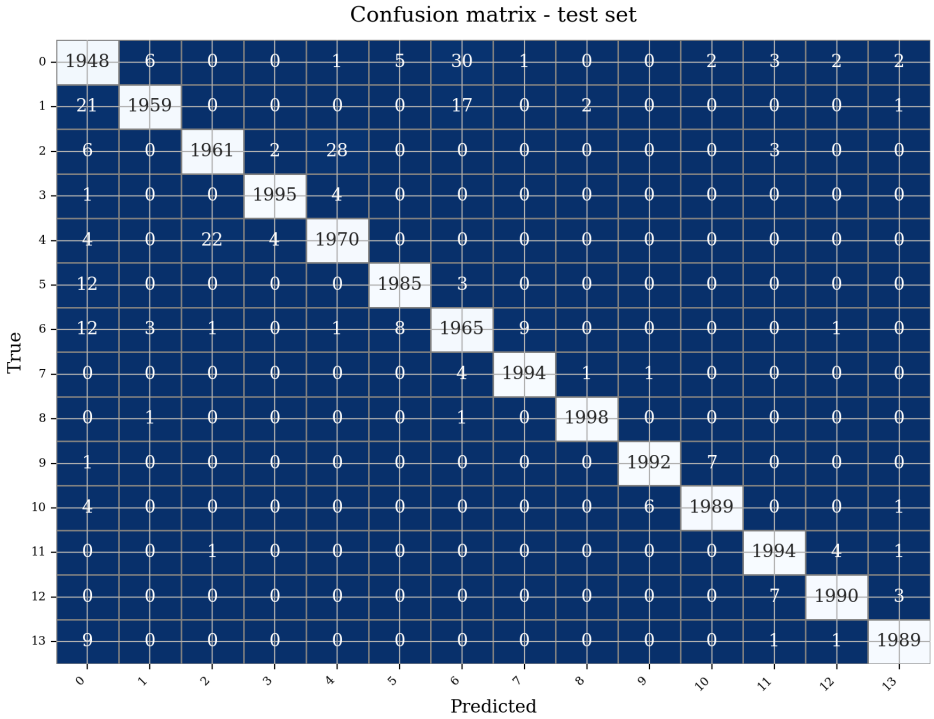

Figure 6: Confusion matrix on the test split. Labels correspond to the fourteen DBpedia-14 categories.

6 Per-Class Metrics and Interpretation

Per-class precision, recall, and F1 scores were computed from the test set. The macro-averaged values for precision, recall, and F1 were 0.9904, 0.9903, and 0.9903, respectively, matching the overall test accuracy. Eleven of fourteen categories achieved F1 scores above 0.99. The lowest F1 scores were recorded for the classes Company at 0.9696 and Building at 0.9776, where there was semantic overlap leading to residual confusion.

Classes Film, Album, and Athlete obtained the highest precision values (≥ 0.996), while NaturalPlace and Village achieved recall values of 0.9970 and 0.9990, respectively. The minimal variation across classes indicates consistent generalization and no overfitting to dominant categories.

Circling back to the attention visualization, the results found in Section 4 supports these quantitative results. The shift from locally diagonal attention to semantically focused patterns after fine-tuning highlights a greater emphasis on contextually relevant tokens such as robot, human, and harm. As previously mentioned, Clark et al. [4] showed that mid-layer attention heads in BERT capture syntactic relations (subject-verb and object-verb dependencies) with accuracies exceeding 75%. Layer 3 in the model represents the analogous middle depth where syntactic and semantic features interact, explaining the observed transition toward content-bearing words after fine-tuning.

This outcome is consistent with transformer interpretability studies. Hewitt and Manning [5] found that BERT's embeddings encode syntactic tree distances with a UUAS of 80. Tenney et al. [6] found that intermediate layers encode syntactic information, while deeper layers capture semantics. Voita et al. [7] further showed that over 40% of attention heads can be pruned with minimal performance loss, indicating specialization among a subset of heads. Together, these findings align with the attention redistribution observed in the model, where the middle layers integrate the structural and semantic features most relevant to the classification objective.

Overall, the combination of uniform per-class metrics, a macro-average F1 score of 0.9903, and consistent attention adaptation supports the conclusion that the fine-tuned model effectively captured domain-specific semantics while maintaining balanced performance across all categories.

Bibliography

[1] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: Smaller, faster, cheaper and lighter," *arXiv preprint arXiv:1910.01108*, 2019.

[2] H. Face, *Distilbert — transformers documentation*, `https://huggingface.co/docs/transformers/en/model_doc/distilbert`, Accessed: October 29, 2025, 2025.

[3] Z. Sun, A. Harit, and P. Lio, *Actionable interpretability via causal hypergraphs: Unravelling batch size effects in deep learning*, 2025. arXiv: `2506.17826 [cs.LG]`. [Online]. Available: `https://arxiv.org/abs/2506.17826`.

[4] K. Clark, U. Khandelwal, O. Levy, and C. D. Manning, "What does bert look at? an analysis of bert's attention," in *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, Florence, Italy: Association for Computational Linguistics, 2019, pp. 276–286.

[5] J. Hewitt and C. D. Manning, "A structural probe for finding syntax in word representations," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, J. Burstein, C. Doran, and T. Solorio, Eds., Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4129–4138. DOI: `10.18653/v1/N19-1419`. [Online]. Available: `https://aclanthology.org/N19-1419/`.

[6] I. Tenney, D. Das, and E. Pavlick, "BERT rediscovers the classical NLP pipeline," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, A. Korhonen, D. Traum, and L. Màrquez, Eds., Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 4593–4601. DOI: `10.18653/v1/P19-1452`. [Online]. Available: `https://aclanthology.org/P19-1452/`.

[7] E. Voita, D. Talbot, F. Moiseev, R. Sennrich, and I. Titov, "Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, A. Korhonen, D. Traum, and L. Màrquez, Eds., Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 5797–5808. DOI: `10.18653/v1/P19-1580`. [Online]. Available: `https://aclanthology.org/P19-1580/`.