

¹ LabelFusion: Learning to Fuse LLMs and Transformer Classifiers for Robust Text Classification

³ Michael Schlee¹, Christoph Weisser¹, Timo Kivimäki², Melchy Mashiku⁴, and
⁴ Benjamin Saefken³

⁵ 1 Centre for Statistics, Georg-August-Universität Göttingen, Germany ² Department of Politics and
⁶ International Studies, University of Bath, Bath, UK ³ Institute of Mathematics, Clausthal University of
⁷ Technology, Clausthal-Zellerfeld, Germany ⁴ Unknown

DOI: [10.xxxxxx/draft](https://doi.org/10.xxxxxx/draft)

Software

- [Review](#) ↗
- [Repository](#) ↗
- [Archive](#) ↗

Editor: [Open Journals](#) ↗

Reviewers:

- [@openjournals](#)

Submitted: 01 January 1970

Published: unpublished

License

Authors of papers retain copyright
and release the work under a

Creative Commons Attribution 4.0
International License ([CC BY 4.0](#)).

⁸ Summary

⁹ LabelFusion is a fusion ensemble for text classification that learns to combine a traditional
¹⁰ transformer-based classifier (e.g., RoBERTa) with one or more Large Language Models (LLMs)
¹¹ such as OpenAI GPT, Google Gemini, or DeepSeek to deliver accurate and cost-aware predictions
¹² across multi-class and multi-label tasks. The package provides a simple high-level interface
¹³ (AutoFusionClassifier) that trains the full pipeline end-to-end with minimal configuration,
¹⁴ and a flexible API for advanced users. Under the hood, LabelFusion concatenates vector signals
¹⁵ from the ML backbone (logits) and LLM(s) (per-class scores) and trains a compact multi-layer
¹⁶ perceptron (FusionMLP) to produce the final prediction. This learned fusion approach captures
¹⁷ complementary strengths of LLM reasoning and traditional transformer-based classifiers,
¹⁸ yielding robust performance across domains—achieving 92.4% accuracy on AG News topic
¹⁹ classification—while enabling practical trade-offs between accuracy, latency, and cost.

²⁰ Statement of Need

²¹ Modern text classification spans diverse scenarios—from sentiment analysis to complex topic
²² tagging—often under constraints that vary per deployment (throughput, cost ceilings, data
²³ privacy). While transformer classifiers such as BERT/RoBERTa achieve strong supervised
²⁴ performance ([Devlin et al., 2018](#); [Liu et al., 2019](#)), frontier LLMs can excel in low-data,
²⁵ ambiguous, or cross-domain settings ([OpenAI, 2023](#)). No single model family is typically
²⁶ uniformly best: LLMs are powerful, but comparatively costly, whereas fine-tuned transformers
²⁷ are efficient but may struggle with out-of-distribution cases.

²⁸ LabelFusion addresses this gap by: (1) exposing a minimal “AutoFusion” interface that trains a
²⁹ learned combination of an ML backbone and one or more LLMs; (2) supporting both multi-class
³⁰ and multi-label classification; (3) providing a lightweight fusion learner that directly fits on LLM
³¹ scores and ML logits; and (4) integrating cleanly with existing ensemble utilities. Researchers
³² and practitioners can therefore leverage LLMs where they add value while retaining the speed
³³ and determinism of transformer models.

³⁴ State of the Field

³⁵ In applied NLP, common tools such as scikit-learn ([Pedregosa et al., 2011](#)) and Hugging Face
³⁶ Transformers ([Wolf et al., 2019](#)) offer strong baselines but do not provide a learned fusion of
³⁷ LLMs with supervised transformers. Orchestration frameworks (e.g., LangChain) focus on tool
³⁸ use rather than classification ensembles. LabelFusion contributes a focused, production-minded
³⁹ implementation of a small learned combiner that operates on per-class signals from both model
⁴⁰ families.

41 Functionality and Design

42 LabelFusion consists of three layers:

- 43 ■ ML component: a RoBERTa-style classifier produces per-class logits for input texts.
- 44 ■ LLM component(s): provider-specific classifiers (OpenAI, Gemini, DeepSeek) return
- 45 per-class scores via prompting. Scores can be cached to minimize API calls when cache
- 46 locations are provided.
- 47 ■ Fusion component: a compact MLP concatenates ML logits and LLM scores and outputs
- 48 fused logits. The ML backbone is trained/fine-tuned with a small learning rate; the fusion
- 49 MLP uses a higher rate, enabling rapid adaptation without destabilizing the encoder.

50 Key features:

- 51 ■ **Multi-class and multi-label support** with consistent data structures and unified training
- 52 pipeline.
- 53 ■ **Optional LLM response caching** reuses on-disk predictions when cache paths are supplied,
- 54 with dataset-hash validation to guard against stale files.
- 55 ■ **Batched scoring** processes multiple texts efficiently with configurable batch sizes for both
- 56 ML tokenization and LLM API calls.
- 57 ■ **Results management** via ResultsManager tracks experiments, stores predictions, com-
- 58 putes metrics, and enables reproducible research workflows.
- 59 ■ **Flexible interfaces**: Command-line training via `train_fusion.py` with YAML configs for
- 60 research; or minimal AutoFusion API for quick deployment.
- 61 ■ **Composable design**: LabelFusion can serve as a strong base learner in higher-level
- 62 ensembles (e.g., voting/weighted combinations of multiple fusion models).

63 Formally, multi-class classification assigns each input $x \in \mathcal{X}$ to exactly one label among K
 64 mutually exclusive classes:

$$f_{\text{mc}} : \mathcal{X} \rightarrow \{1, \dots, K\}.$$

65 In contrast, multi-label classification predicts a subset of relevant classes, represented as a
 66 binary indicator vector $\mathbf{y} \in \{0, 1\}^K$, where $y_k = 1$ denotes membership in class k :

$$f_{\text{ml}} : \mathcal{X} \rightarrow \{0, 1\}^K.$$

67 Minimal Example (AutoFusion)

```
from textclassify import AutoFusionClassifier

config = {
    'llm_provider': 'deepeek',
    'label_columns': ['positive', 'negative', 'neutral']
}

clf = AutoFusionClassifier(config)
clf.fit(train_dataframe)           # trains ML backbone, gathers LLM scores, fits fusi
pred = clf.predict(["This is amazing!"]) # fused prediction
```

68 CLI and Configuration

69 Users can generate a starter config and train via the command line:

- 70 ■ Create config: `python train_fusion.py --create-config fusion_config.yaml`
- 71 ■ Train: `python train_fusion.py --config fusion_config.yaml`
- 72 ■ Optional test data and output artifacts are also supported.

73 Quality Control

74 The repository ships legacy unit tests under `tests/evaluation/old/` that cover configuration
 75 handling, core types, and package integration. Fusion-specific logic is currently exercised
 76 through CLI-driven workflows and notebooks that run end-to-end training with deterministic
 77 seeds where applicable.

78 Evaluation scripts (`tests/evaluation/`) provide comprehensive benchmarking on standard
 79 datasets: - **AG News** (Zhang et al., 2015): 4-class topic classification with experiments
 80 across varying training data sizes (20%–100%) - **GoEmotions** (Demszky et al., 2020): 28-class
 81 multi-label emotion classification for validating multi-label fusion performance

82 LLM scoring paths implement retries and disk caching; transformer training supports standard
 83 sanity checks (overfit a small batch, reduced batch sizes for constrained hardware). Metrics
 84 (accuracy/F1, per-label scores) are computed automatically and stored with run artifacts to
 85 facilitate regression tracking and reproducibility.

86 Availability and Installation

87 LabelFusion is distributed as part of the `textclassify` package under the MIT license and
 88 is available at <https://github.com/DataandAIResearch/LabelFusion>. The fusion components
 89 require Python 3.8+ and common scientific Python dependencies (PyTorch, transformers,
 90 scikit-learn, numpy, pandas, PyYAML). Optional plotting depends on matplotlib/seaborn.
 91 Installation and quick-start snippets are provided in the README and FUSION_README.md.

92 Production-Ready Features

93 Beyond the core fusion methodology, LabelFusion includes features for practical deployment:

- 94 **▪ LLM Response Caching:** Optional disk-backed caches reuse prior predictions when cache
 95 paths are supplied, with dataset hashes to flag inconsistent inputs.
- 96 **▪ Results Management:** Built-in `ResultsManager` tracks experiments, stores predictions,
 97 and computes metrics automatically. Supports comparison across runs and configuration
 98 tracking.
- 99 **▪ Batch Processing:** Efficient batched scoring of texts with configurable batch sizes for
 100 both ML and LLM components.

101 Impact and Use Cases

102 Empirical Performance

103 LabelFusion has been evaluated on standard benchmark datasets to validate its effectiveness.
 104 Key findings demonstrate consistent improvements over individual model components:

105 AG News Topic Classification

106 Evaluation on the AG News dataset (Zhang et al., 2015) (4-class topic classification) with
 107 5,000 test samples shows:

Training Data	Model	Accuracy	F1-Score	Precision	Recall
20% (800)	Fusion	92.2%	0.922	0.923	0.922
20% (800)	RoBERTa	89.8%	0.899	0.902	0.898
20% (800)	OpenAI	84.4%	0.844	0.857	0.844
40% (1,600)	Fusion	92.2%	0.922	0.924	0.922
40% (1,600)	RoBERTa	91.0%	0.911	0.913	0.910
40% (1,600)	OpenAI	84.4%	0.844	0.857	0.844
100% (4,000)	Fusion	92.4%	0.924	0.926	0.924

Training Data	Model	Accuracy	F1-Score	Precision	Recall
100% (4,000)	RoBERTa	92.2%	0.922	0.923	0.922
100% (4,000)	OpenAI	84.4%	0.844	0.857	0.844

108 **Key Observations:** - Fusion consistently outperforms individual models across all training
 109 data sizes - With only 20% training data, Fusion achieves 92.2% accuracy—matching its
 110 performance with full data - Demonstrates superior **data efficiency**: fusion learning extracts
 111 maximum value from limited examples - RoBERTa alone requires 100% of data to approach
 112 Fusion's 20% performance - LLM (OpenAI) shows stable but lower performance, highlighting
 113 the value of combining approaches

114 These results validate that learned fusion captures complementary strengths: the LLM provides
 115 robust reasoning even with limited training data, while the ML backbone adds efficiency and
 116 domain-specific patterns.

117 Application Domains

118 Learned fusion excels in scenarios where model strengths complement each other:

- 119 ▪ **Customer feedback analysis** with nuanced multi-label taxonomies where LLMs handle
 120 ambiguous sentiment while ML models efficiently process clear cases
- 121 ▪ **Content moderation** where uncertain cases benefit from LLM reasoning while rou-
 122 tine items rely on the fast ML backbone, enabling real-time processing with accuracy
 123 guarantees
- 124 ▪ **Scientific literature classification** across heterogeneous topics where domain shift is
 125 common and LLMs provide robustness to new terminology
- 126 ▪ **Low-resource settings** where limited training data is available but task complexity requires
 127 sophisticated reasoning

128 The approach enables pragmatic cost control (e.g., the fusion layer learns when to rely more
 129 heavily on the efficient ML backbone versus the more expensive LLM signal) while retaining a
 130 single trainable decision surface that optimizes for the specific deployment constraints.

131 Acknowledgements

132 We thank contributors and users who reported issues and shared datasets. LabelFusion
 133 builds on the open-source ecosystem, notably Hugging Face Transformers ([Wolf et al., 2019](#)),
 134 scikit-learn ([Pedregosa et al., 2011](#)), PyTorch ([Paszke et al., 2019](#)), and LLM provider SDKs.
 135 We acknowledge the use of the AG News and GoEmotions benchmark datasets for evaluation.

136 References

- 137 Demszky, D., Movshovitz-Attias, D., Ko, J., Cowen, A., Nemade, G., & Ravi, S. (2020).
 138 GoEmotions: A dataset of fine-grained emotions. *Proceedings of the 58th Annual Meeting*
 139 *of the Association for Computational Linguistics*, 4040–4054.
- 140 Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of deep
 141 bidirectional transformers for language understanding. *arXiv Preprint arXiv:1810.04805*.
- 142 Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer,
 143 L., & Stoyanov, V. (2019). RoBERTa: A robustly optimized BERT pretraining approach.
 144 *arXiv Preprint arXiv:1907.11692*.
- 145 OpenAI. (2023). GPT-4 technical report. *arXiv Preprint arXiv:2303.08774*.
- 146 Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin,
 147 Z., Gimelshein, N., Antiga, L., & others. (2019). PyTorch: An imperative style, high-

- 148 performance deep learning library. *Advances in Neural Information Processing Systems*,
149 32.
- 150 Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M.,
151 Prettenhofer, P., Weiss, R., Dubourg, V., & others. (2011). Scikit-learn: Machine learning
152 in python. *Journal of Machine Learning Research*, 12(Oct), 2825–2830.
- 153 Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T.,
154 Louf, R., Funtowicz, M., & others. (2019). HuggingFace's transformers: State-of-the-art
155 natural language processing. *arXiv Preprint arXiv:1910.03771*.
- 156 Zhang, X., Zhao, J., & LeCun, Y. (2015). Character-level convolutional networks for text
157 classification. *Advances in Neural Information Processing Systems*, 28, 649–657.

DRAFT