

# <sup>1</sup> LabelFusion: Learning to Fuse LLMs and Transformer Classifiers for Robust Text Classification

<sup>3</sup> Michael Schlee<sup>1</sup>, Christoph Weisser<sup>1</sup>, Timo Kivimäki<sup>2</sup>, Melchizedek  
<sup>4</sup> Mashiku<sup>4</sup>, and Benjamin Saefken<sup>3</sup>

<sup>5</sup> 1 Centre for Statistics, Georg-August-Universität Göttingen, Germany <sup>2</sup> Department of Politics and  
<sup>6</sup> International Studies, University of Bath, Bath, UK <sup>3</sup> Institute of Mathematics, Clausthal University of  
<sup>7</sup> Technology, Clausthal-Zellerfeld, Germany <sup>4</sup> Tanaq Management Services LLC, Contracting Agency to  
<sup>8</sup> the Division of Viral Diseases, Centers for Disease Control and Prevention, Chamblee, Georgia, USA

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Published: unpublished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](#)).  
21  
22  
23

## <sup>9</sup> Summary

<sup>10</sup> LabelFusion is a fusion ensemble for text classification that learns to combine a traditional  
<sup>11</sup> transformer-based classifier (e.g., RoBERTa) with one or more Large Language Models (LLMs)  
<sup>12</sup> such as OpenAI GPT, Google Gemini, or DeepSeek to deliver accurate and cost-aware predictions  
<sup>13</sup> across multi-class and multi-label tasks. The package provides a simple high-level interface  
<sup>14</sup> (AutoFusionClassifier) that trains the full pipeline end-to-end with minimal configuration,  
<sup>15</sup> and a flexible API for advanced users. Under the hood, LabelFusion concatenates vector signals  
<sup>16</sup> from the ML backbone (logits) and LLM(s) (per-class scores) and trains a compact multi-layer  
<sup>17</sup> perceptron (FusionMLP) to produce the final prediction. This learned fusion approach captures  
<sup>18</sup> complementary strengths of LLM reasoning and traditional transformer-based classifiers,  
<sup>19</sup> yielding robust performance across domains—achieving 92.4% accuracy on AG News topic  
<sup>20</sup> classification—while enabling practical trade-offs between accuracy, latency, and cost.

## <sup>21</sup> Statement of Need

<sup>22</sup> Modern text classification spans diverse scenarios, from sentiment analysis (Kant et al., 2024;  
Luber et al., 2021; Thormann et al., 2021) to complex topic tagging (Kant et al., 2022; A.  
Thielmann, Weisser, Krenz, & Säfken, 2021; A. Thielmann, Weisser, & Krenz, 2021; A. F.  
Thielmann et al., 2024), often under constraints that vary per deployment (throughput, cost  
ceilings, data privacy). While transformer classifiers such as BERT/RoBERTa achieve strong  
supervised performance (Devlin et al., 2018; Liu et al., 2019), frontier LLMs can excel in  
low-data, ambiguous, or cross-domain settings (OpenAI, 2023). No single model family is  
typically uniformly best: LLMs are powerful, but comparatively costly, whereas fine-tuned  
transformers are efficient but may struggle with out-of-distribution cases.

<sup>31</sup> LabelFusion addresses this gap by: (1) exposing a minimal “AutoFusion” interface that trains a  
<sup>32</sup> learned combination of an ML backbone and one or more LLMs; (2) supporting both multi-class  
<sup>33</sup> and multi-label classification; (3) providing a lightweight fusion learner that directly fits on LLM  
<sup>34</sup> scores and ML logits; and (4) integrating cleanly with existing ensemble utilities. Researchers  
<sup>35</sup> and practitioners can therefore leverage LLMs where they add value while retaining the speed  
<sup>36</sup> and determinism of transformer models.

## <sup>37</sup> State of the Field

<sup>38</sup> In applied NLP, common tools such as scikit-learn (Pedregosa et al., 2011) and Hugging Face  
<sup>39</sup> Transformers (Wolf et al., 2019) offer strong baselines but do not provide a learned fusion of  
<sup>40</sup> LLMs with supervised transformers. Orchestration frameworks (e.g., LangChain) focus on tool  
<sup>41</sup> use rather than classification ensembles. LabelFusion contributes a focused, production-minded

<sup>42</sup> implementation of a small learned combiner that operates on per-class signals from both model  
<sup>43</sup> families.

## <sup>44</sup> Functionality and Design

<sup>45</sup> LabelFusion consists of three layers:

- <sup>46</sup> ▪ ML component: a RoBERTa-style classifier produces per-class logits for input texts.
- <sup>47</sup> ▪ LLM component(s): provider-specific classifiers (OpenAI, Gemini, DeepSeek) return
- <sup>48</sup> per-class scores via prompting. Scores can be cached to minimize API calls when cache
- <sup>49</sup> locations are provided.
- <sup>50</sup> ▪ Fusion component: a compact MLP concatenates ML logits and LLM scores and outputs
- <sup>51</sup> fused logits. The ML backbone is trained/fine-tuned with a small learning rate; the fusion
- <sup>52</sup> MLP uses a higher rate, enabling rapid adaptation without destabilizing the encoder.

<sup>53</sup> Key features:

- <sup>54</sup> ▪ **Multi-class and multi-label support** with consistent data structures and unified training
- <sup>55</sup> pipeline.
- <sup>56</sup> ▪ **Optional LLM response caching** reuses on-disk predictions when cache paths are supplied,
- <sup>57</sup> with dataset-hash validation to guard against stale files.
- <sup>58</sup> ▪ **Batched scoring** processes multiple texts efficiently with configurable batch sizes for both
- <sup>59</sup> ML tokenization and LLM API calls.
- <sup>60</sup> ▪ **Results management** via ResultsManager tracks experiments, stores predictions, com-
- <sup>61</sup> putes metrics, and enables reproducible research workflows.
- <sup>62</sup> ▪ **Flexible interfaces**: Command-line training via `train_fusion.py` with YAML configs for
- <sup>63</sup> research; or minimal AutoFusion API for quick deployment.
- <sup>64</sup> ▪ **Composable design**: LabelFusion can serve as a strong base learner in higher-level
- <sup>65</sup> ensembles (e.g., voting/weighted combinations of multiple fusion models).

<sup>66</sup> Formally, multi-class classification assigns each input  $x \in \mathcal{X}$  to exactly one label among  $K$   
<sup>67</sup> mutually exclusive classes:

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

<sup>68</sup> In contrast, multi-label classification predicts a subset of relevant classes, represented as a  
<sup>69</sup> binary indicator vector  $y \in \{0, 1\}^K$ , where  $y_k = 1$  denotes membership in class  $k$ :

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

## <sup>70</sup> Minimal Example (AutoFusion)

```
from textclassify import AutoFusionClassifier

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

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

## <sup>71</sup> CLI and Configuration

<sup>72</sup> Users can generate a starter config and train via the command line:

- <sup>73</sup> ▪ Create config: `python train_fusion.py --create-config fusion_config.yaml`
- <sup>74</sup> ▪ Train: `python train_fusion.py --config fusion_config.yaml`
- <sup>75</sup> ▪ Optional test data and output artifacts are also supported.

## 76 Quality Control

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

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

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

## 89 Availability and Installation

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

## 95 Production-Ready Features

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

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

## 104 Impact and Use Cases

### 105 Empirical Performance

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

#### 108 AG News Topic Classification

109 Evaluation on the AG News dataset (Zhang et al., 2015) (4-class topic classification) with  
 110 5,000 test samples shows:

Training Data	Model	Accuracy	F1-Score	Precision	Recall
20% (800)	<b>Fusion</b>	<b>92.2%</b>	<b>0.922</b>	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)	<b>Fusion</b>	<b>92.2%</b>	<b>0.922</b>	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)	<b>Fusion</b>	<b>92.4%</b>	<b>0.924</b>	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

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

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

## 120 Application Domains

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

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

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

## 134 Acknowledgements

135 We thank contributors and users who reported issues and shared datasets. LabelFusion builds on  
 136 the open-source ecosystem, notably Hugging Face Transformers ([Wolf et al., 2019](#)), scikit-learn  
 137 ([Pedregosa et al., 2011](#)), PyTorch ([Paszke et al., 2019](#)), and LLM provider SDKs. The work  
 138 presented in this paper was conducted independently by the author Melchizedek Mashiku and  
 139 is not affiliated with Tanaq Management Services LLC, Contracting Agency to the Division  
 140 of Viral Diseases, Centers for Disease Control and Prevention, Chamblee, Georgia, USA. We  
 141 acknowledge the use of the AG News and GoEmotions benchmark datasets for evaluation.

## 142 References

- 143 Demszky, D., Movshovitz-Attias, D., Ko, J., Cowen, A., Nemade, G., & Ravi, S. (2020).  
 144 GoEmotions: A dataset of fine-grained emotions. *Proceedings of the 58th Annual Meeting*  
 145 *of the Association for Computational Linguistics*, 4040–4054.
- 146 Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of deep  
 147 bidirectional transformers for language understanding. *arXiv Preprint arXiv:1810.04805*.  
 148 <https://doi.org/10.48550/arXiv.1810.04805>
- 149 Kant, G., Wiebelt, L., Weisser, C., Kis-Katos, K., Luber, M., & Säfken, B. (2022). An  
 150 iterative topic model filtering framework for short and noisy user-generated data: Analyzing  
 151 conspiracy theories on twitter. *International Journal of Data Science and Analytics*, 20(2),  
 152 269–289. <https://doi.org/10.1007/s41060-022-00321-4>

- 153 Kant, G., Zhelyazkov, I., Thielmann, A., Weisser, C., Schlee, M., Ehrling, C., Säfken, B., &  
 154 Kneib, T. (2024). One-way ticket to the moon? An NLP-based insight on the phenomenon  
 155 of small-scale neo-broker trading. *Social Network Analysis and Mining*, 14(1), 121. <https://doi.org/10.1007/s13278-024-01273-2>
- 157 Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer,  
 158 L., & Stoyanov, V. (2019). RoBERTa: A robustly optimized BERT pretraining approach.  
 159 *arXiv Preprint arXiv:1907.11692*. <https://doi.org/10.48550/arXiv.1907.11692>
- 160 Luber, M., Weisser, C., Säfken, B., Silbersdorff, A., Kneib, T., & Kis-Katos, K. (2021).  
 161 Identifying topical shifts in twitter streams: An integration of non-negative matrix factori-  
 162 sation, sentiment analysis and structural break models for large scale data. In J. Bright, A.  
 163 Giachanou, V. Spaiser, F. Spezzano, A. George, & A. Pavliuc (Eds.), *Disinformation in open  
 164 online media* (pp. 33–49). Springer International Publishing. ISBN: 978-3-030-87031-7
- 165 OpenAI. (2023). GPT-4 technical report. *arXiv Preprint arXiv:2303.08774*. <https://doi.org/10.48550/arXiv.2303.08774>
- 166 Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T.,  
 167 Lin, Z., Gimelshein, N., Antiga, L., & others. (2019). PyTorch: An impera-  
 168 tive style, high-performance deep learning library. *Advances in Neural Informa-  
 169 tion Processing Systems*, 32. <https://proceedings.neurips.cc/paper/2019/hash/bdbca288fee7f92f2bfa9f7012727740-Abstract.html>
- 172 Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel,  
 173 M., Prettenhofer, P., Weiss, R., Dubourg, V., & others. (2011). Scikit-learn: Machine  
 174 learning in python. *Journal of Machine Learning Research*, 12(Oct), 2825–2830. <http://www.jmlr.org/papers/v12/pedregosa11a.html>
- 176 Thielmann, A. F., Weisser, C., & Säfken, B. (2024). Human in the loop: How to effectively  
 177 create coherent topics by manually labeling only a few documents per class. In N. Calzolari,  
 178 M.-Y. Kan, V. Hoste, A. Lenci, S. Sakti, & N. Xue (Eds.), *Proceedings of the 2024 joint  
 179 international conference on computational linguistics, language resources and evaluation  
 180 (LREC-COLING 2024)* (pp. 8395–8405). ELRA; ICCL. <https://aclanthology.org/2024.lrec-main.736/>
- 182 Thielmann, A., Weisser, C., & Krenz, A. (2021). One-class support vector machine and LDA  
 183 topic model integration—evidence for AI patents. In N. H. Phuong & V. Kreinovich (Eds.),  
 184 *Soft computing: Biomedical and related applications* (pp. 263–272). Springer International  
 185 Publishing. [https://doi.org/10.1007/978-3-030-76620-7\\_23](https://doi.org/10.1007/978-3-030-76620-7_23)
- 186 Thielmann, A., Weisser, C., Krenz, A., & Säfken, B. (2021). Unsupervised document  
 187 classification integrating web scraping, one-class SVM and LDA topic modelling. *Journal  
 188 of Applied Statistics*, 50(3), 574–591. <https://doi.org/10.1080/02664763.2021.1919063>
- 189 Thormann, M.-L., Farchmin, J., Weisser, C., Kruse, R.-M., Säfken, B., & Silbersdorff, A.  
 190 (2021). Stock price predictions with LSTM neural networks and twitter sentiment. *Statistics,  
 191 Optimization & Information Computing*, 9(2), 268–287. <https://doi.org/10.19139/soic-2310-5070-1202>
- 193 Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T.,  
 194 Louf, R., Funtowicz, M., & others. (2019). HuggingFace's transformers: State-of-the-art  
 195 natural language processing. *arXiv Preprint arXiv:1910.03771*. <https://doi.org/10.48550/arXiv.1910.03771>
- 197 Zhang, X., Zhao, J., & LeCun, Y. (2015). Character-level convolutional networks for text  
 198 classification. *Advances in Neural Information Processing Systems*, 28, 649–657. <https://papers.nips.cc/paper/2015/hash/250cf8b51c773f3f8dc8b4be867a9a02-Abstract.html>