

# <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Published: unpublished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](#))<sup>3</sup>

## <sup>9</sup> Summary

<sup>10</sup> LabelFusion is a novel 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  
<sup>13</sup> predictions across multi-class and multi-label tasks. The package provides a simple high-level  
<sup>14</sup> interface (AutoFusionClassifier) that trains the full pipeline end-to-end with minimal  
<sup>15</sup> configuration, and a flexible API for advanced users. Under the hood, LabelFusion integrates  
<sup>16</sup> vector signals from both sources by concatenating the ML backbone's embeddings with the  
<sup>17</sup> LLM-derived per-class scores—obtained through structured prompt-engineering strategies—and  
<sup>18</sup> feeds this joint representation into a compact multi-layer perceptron (FusionMLP) that produces  
<sup>19</sup> the final prediction. This learned fusion approach captures complementary strengths of LLM  
<sup>20</sup> reasoning and traditional transformer-based classifiers, yielding robust performance across  
<sup>21</sup> domains—achieving 92.4% accuracy on AG News and 92.3% on 10-class Reuters 21578 topic  
<sup>22</sup> classification — while enabling practical trade-offs between accuracy, latency, and cost.

## Statement of Need

<sup>24</sup> Modern text classification spans diverse scenarios, from sentiment analysis ([Kant et al., 2024](#);  
<sup>25</sup> [Luber et al., 2021](#); [Thormann et al., 2021](#)) to complex topic tagging ([Kant et al., 2022](#); [A.  
26 Thielmann, Weisser, Krenz, & Säfken, 2021](#); [A. Thielmann, Weisser, & Krenz, 2021](#); [A. F.  
27 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 or extremely limited  
training examples.

<sup>34</sup> LabelFusion addresses this gap by: (1) exposing a minimal “AutoFusion” interface that trains a  
<sup>35</sup> learned combination of an ML backbone and one or more LLMs; (2) supporting both multi-class  
<sup>36</sup> and multi-label classification; (3) providing a lightweight fusion learner that directly fits on  
<sup>37</sup> LLM scores and ML embeddings; and (4) integrating cleanly with existing ensemble utilities.  
<sup>38</sup> Researchers and practitioners can therefore leverage LLMs where they add value while retaining  
<sup>39</sup> the speed and determinism of transformer models.

## 40 State of the Field

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

## 47 Software design

48 The design of LabelFusion is based on three core principles: **modularity**, **composability**, and  
49 **reproducibility**. This is achieved through a consistent object-oriented API that unifies disparate  
50 model types—traditional machine learning (ML) and Large Language Models (LLMs)—under  
51 a common interface. All classifiers inherit from the `BaseClassifier` abstract base class, which  
52 standardizes the `predict()` interface and the `ClassificationResult` data structure. The  
53 `BaseLLMClassifier` further extends `AsyncBaseClassifier` to manage the latency of API calls.  
54 Within the LLM module, the `PromptEngineer` dynamically constructs context-aware instructions  
55 and classification guidelines based on the label schema and training examples, ensuring the  
56 LLM produces semantically aligned per-class scores. The ML module, exemplified by the  
57 `RoBERTaClassifier`, extracts the **768-dimensional [CLS] token embeddings**, which serve as a  
58 crucial input signal for the fusion component.  
59 The core of the fusion module is the **Multi-Layer Perceptron (FusionMLP)**, implemented as  
60 a `torch.nn.Module`. The FusionMLP accepts a **concatenated input vector** combining the  
61 ML embeddings and the LLM's per-class scores ( $768 + K$  dimensions). Training employs a  
62 **differential learning rate strategy**: the ML backbone is fine-tuned with a low rate ( $10^{-5}$ ),  
63 while the FusionMLP head is trained with a higher rate ( $10^{-3}$ ) for rapid adaptation. The  
64 `AutoFusionClassifier` abstracts this entire orchestration behind a single `fit()` interface,  
65 prioritizing usability and reproducibility while retaining access to lower-level components.

## 66 Research Impact Statement

67 LabelFusion implements a learned fusion architecture for text classification that combines a  
68 supervised transformer-based classifier (e.g., RoBERTa) with one or more Large Language  
69 Models (LLMs such as OpenAI GPT, Google Gemini, or DeepSeek). The core design follows a  
70 two-stage ensemble approach: the transformer backbone produces embedding- and logit-level  
71 signals, while the LLM component generates per-class scores via prompt-based classification.  
72 These heterogeneous signals are concatenated and passed to a compact fusion multi-layer  
73 perceptron (FusionMLP) that learns how to combine both sources into a single prediction.  
74 A key component of the design is the integrated prompt engineering module for the LLM  
75 classifiers. Rather than relying on static prompts, LabelFusion includes a prompt warehouse  
76 that dynamically constructs task-specific instructions based on the provided label schema and  
77 training examples. This mechanism generates context-aware role descriptions and classification  
78 guidelines, ensuring that the LLM produces consistent and semantically aligned per-class scores  
79 for downstream fusion. The software is organized into three clearly separated components: an  
80 ML classifier module, LLM classifier modules, and a fusion module. A central design decision is  
81 the high-level `AutoFusionClassifier`, which abstracts orchestration of the full pipeline—including  
82 data splitting, automated prompt generation, optional caching of LLM predictions, and  
83 fusion model training—behind a single `fit()` interface. This design prioritizes usability and  
84 reproducibility while retaining access to lower-level components for custom workflows and  
85 experimentation.

## 86 Functionality and Design

87 LabelFusion consists of three layers:

- 88     ▪ ML component: a RoBERTa-style classifier produces per-class logits for input texts.
- 89     ▪ LLM component(s): provider-specific classifiers (OpenAI, Gemini, DeepSeek) return
- 90       per-class scores. Scores can be cached to minimize API calls when cache locations are
- 91       provided.
- 92     ▪ Fusion component: a compact MLP concatenates information rich ML embeddings and
- 93       LLM scores and outputs fused logits. The ML backbone is trained/fine-tuned with a
- 94       small learning rate; the fusion MLP uses a higher rate, enabling rapid adaptation without
- 95       destabilizing the encoder.

96     Key features:

- 97       ▪ **Multi-class and multi-label support** with consistent data structures and unified training
- 98       pipeline.
- 99       ▪ **Optional LLM response caching** reuses on-disk predictions when cache paths are supplied,
- 100       with dataset-hash validation to guard against stale files.
- 101       ▪ **Batched scoring** processes multiple texts efficiently with configurable batch sizes for both
- 102       ML tokenization and LLM API calls.
- 103       ▪ **Results management** via ResultsManager tracks experiments, stores predictions,
- 104       computes metrics, and enables reproducible research workflows.
- 105       ▪ **Flexible interfaces**: Command-line training via `train_fusion.py` with YAML configs for
- 106       research; or minimal AutoFusion API for quick deployment.
- 107       ▪ **Composable design**: LabelFusion can serve as a strong base learner in higher-level
- 108       ensembles (e.g., voting/weighted combinations of multiple fusion models).

109     We support both multi-class setups (one label per input) and multi-label scenarios (multiple

110       labels per input), and point readers to Appendix A for formal definitions and training

111       implications.

112     Minimal Example (AutoFusion)

```
from textclassify.ensemble.auto_fusion import AutoFusionClassifier

# Multi-class: exactly one of the sentiment labels applies
multiclass_config = {
    'llm_provider': 'deepseek',
    'label_columns': ['positive', 'negative', 'neutral'],
    'multi_label': False
}
multiclass_clf = AutoFusionClassifier(multiclass_config)
multiclass_clf.fit(train_dataframe)
multiclass_pred = multiclass_clf.predict(["This is amazing!"])

# Multi-label: news article can belong to several topics simultaneously
multilabel_config = {
    'llm_provider': 'deepseek',
    'label_columns': ['politics', 'economy', 'technology'],
    'multi_label': True
}
multilabel_clf = AutoFusionClassifier(multilabel_config)
multilabel_clf.fit(train_dataframe)
multilabel_pred = multilabel_clf.predict(["New investment in AI chips"])
```

113     Quality Control

114     The repository ships legacy unit tests under `tests/evaluation/old/` that cover configuration

115       handling, core types, and package integration. Fusion-specific logic is currently exercised

116 through CLI-driven workflows and notebooks that run end-to-end training with deterministic  
 117 seeds where applicable.

118 Evaluation scripts (tests/evaluation/) provide comprehensive benchmarking on standard  
 119 datasets: - **AG News** ([Zhang et al., 2015](#)): 4-class topic classification with experiments  
 120 across varying training data sizes (20%–100%) - **Reuters-21578** ([Lewis, 1997](#)): A single-label  
 121 10-class subset of the Reuters-21578 corpus, used to evaluate multi-class fusion performance  
 122 on moderately imbalanced news topics.

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

## 127 Availability and Installation

128 LabelFusion is distributed as part of the textclassify package under the MIT license and  
 129 is available at <https://github.com/DataandAIResearch/LabelFusion>. The fusion components  
 130 require Python 3.8+ and common scientific Python dependencies (PyTorch, transformers,  
 131 scikit-learn, numpy, pandas, PyYAML, matplotlib, seaborn). Installation and quick-start  
 132 snippets are provided in the README.

## 133 Production-Ready Features

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

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

## 142 Impact and Use Cases

### 143 Empirical Performance

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

146 AG News Topic Classification Evaluation on the AG News dataset ([Zhang et al., 2015](#))  
 147 (4-class topic classification) with 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	85.1%	0.847	0.863	0.846
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	83.9%	0.835	0.847	0.834
60% (2,400)	<b>Fusion</b>	<b>92.0%</b>	<b>0.920</b>	0.922	0.920
60% (2,400)	RoBERTa	91.0%	0.910	0.911	0.910
60% (2,400)	OpenAI	85.2%	0.847	0.861	0.844
80% (3,200)	<b>Fusion</b>	<b>91.6%</b>	<b>0.916</b>	0.917	0.916
80% (3,200)	RoBERTa	91.4%	0.914	0.915	0.914

Training Data	Model	Accuracy	F1-Score	Precision	Recall
80% (3,200)	OpenAI	84.1%	0.837	0.849	0.832
100% (4,000)	<b>Fusion</b>	<b>92.4%</b>	<b>0.924</b>	0.926	0.924
100% (4,000)	RoBERTa	92.2%	0.922	0.923	0.922
100% (4,000)	OpenAI	85.3%	0.849	0.868	0.847

<sup>148</sup> **Key Observations:** - Fusion consistently outperforms individual models across all training  
<sup>149</sup> data sizes - With only 20% training data, Fusion achieves 92.2% accuracy—matching its  
<sup>150</sup> performance with full data - Demonstrates superior **data efficiency**: fusion learning extracts  
<sup>151</sup> maximum value from limited examples - RoBERTa alone requires 100% of data to approach  
<sup>152</sup> Fusion's 20% performance - LLM (OpenAI) shows stable but lower performance, highlighting  
<sup>153</sup> the value of combining approaches

#### <sup>154</sup> Reuters-21578 Topic Classification

Training Data	Model	Accuracy	F1-Score	Precision	Recall
20% (1168)	<b>Fusion</b>	72.0%	0.752	0.769	0.745
20% (1168)	RoBERTa	67.3%	0.534	0.465	0.643
20% (1168)	OpenAI	88.6%	0.928	0.951	0.923
40% (2336)	<b>Fusion</b>	83.6%	0.886	0.893	0.889
40% (2336)	RoBERTa	82.0%	0.836	0.858	0.850
40% (2336)	OpenAI	87.9%	0.931	0.952	0.917
60% (3505)	<b>Fusion</b>	85.5%	0.932	0.929	0.950
60% (3505)	RoBERTa	83.4%	0.907	0.906	0.945
60% (3505)	OpenAI	88.4%	0.938	0.959	0.924
80% (4673)	<b>Fusion</b>	90.2%	0.954	0.954	0.965
80% (4673)	RoBERTa	88.8%	0.943	0.930	0.966
80% (4673)	OpenAI	88.0%	0.934	0.951	0.918
100% (5842)	<b>Fusion</b>	92.3%	0.960	0.967	0.961
100% (5842)	RoBERTa	89.0%	0.946	0.932	0.966
100% (5842)	OpenAI	88.9%	0.939	0.963	0.927

<sup>155</sup> **Key Observations:** - Fusion consistently outperforms individual models across all training  
<sup>156</sup> data sizes - With only 20% training data, Fusion achieves 92.2% accuracy—matching its  
<sup>157</sup> performance with full data - Demonstrates superior **data efficiency**: fusion learning extracts  
<sup>158</sup> maximum value from limited examples - RoBERTa alone requires 100% of data to approach  
<sup>159</sup> Fusion's 20% performance - LLM (OpenAI) shows stable but lower performance, highlighting  
<sup>160</sup> the value of combining approaches

Training Data	Model	Accuracy	F1-Score	Precision	Recall
5% (292)	<b>Fusion</b>	<b>70.6%</b>	<b>0.717</b>	0.720	0.715
5% (292)	RoBERTa	0.0%	0.372	0.276	0.713
5% (292)	OpenAI	88.1%	0.930	0.952	0.917
10% (584)	<b>Fusion</b>	<b>67.0%</b>	<b>0.671</b>	0.672	0.671
10% (584)	RoBERTa	40.0%	0.417	0.321	0.616
10% (584)	OpenAI	88.5%	0.938	0.962	0.926
20% (1168)	<b>Fusion</b>	<b>72.0%</b>	<b>0.752</b>	0.769	0.745
20% (1168)	RoBERTa	67.3%	0.534	0.465	0.643
20% (1168)	OpenAI	88.6%	0.928	0.951	0.923
40% (2336)	<b>Fusion</b>	<b>83.6%</b>	<b>0.886</b>	0.893	0.889
40% (2336)	RoBERTa	82.0%	0.836	0.858	0.850

Training Data	Model	Accuracy	F1-Score	Precision	Recall
40% (2336)	OpenAI	87.9%	0.931	0.952	0.917
60% (3505)	<b>Fusion</b>	<b>85.5%</b>	<b>0.932</b>	0.929	0.950
60% (3505)	RoBERTa	83.4%	0.907	0.906	0.945
60% (3505)	OpenAI	88.4%	0.938	0.959	0.924
80% (4673)	<b>Fusion</b>	<b>90.2%</b>	<b>0.954</b>	0.954	0.965
80% (4673)	RoBERTa	88.8%	0.943	0.930	0.966
80% (4673)	OpenAI	88.0%	0.934	0.951	0.918
100% (5842)	<b>Fusion</b>	<b>92.3%</b>	<b>0.960</b>	0.967	0.961
100% (5842)	RoBERTa	89.0%	0.946	0.932	0.966
100% (5842)	OpenAI	88.9%	0.939	0.963	0.927

161 **Key Observations:** - In extremely low-data settings, the Fusion Ensembles appear negatively  
162 affected by the RoBERTa component, resulting in reduced overall prediction performance -  
163 The LLM (OpenAI) is the preferred model in low-data regimes for multi-label classification on  
164 the 10-class Reuters-21578 subset - RoBERTa alone requires around 80% of the training data  
165 to reach the LLM's performance at only 5% - In high-data settings (80% to 100%), Fusion  
166 Ensembles outperform the individual models by a substantial margin. - The EnsembleFusion  
167 approach attains the best overall prediction performance at 92.3%

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

171 **Application Domains**

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

- 173 ▪ **Customer feedback analysis** with nuanced multi-label taxonomies where LLMs handle  
174 ambiguous sentiment while ML models efficiently process clear cases
- 175 ▪ **Content moderation** where uncertain cases benefit from LLM reasoning while routine  
176 items rely on the fast ML backbone, enabling real-time processing with accuracy  
177 guarantees
- 178 ▪ **Scientific literature classification** across heterogeneous topics where domain shift is  
179 common and LLMs provide robustness to new terminology
- 180 ▪ **Low-resource settings** where limited training data is available but task complexity requires  
181 sophisticated reasoning

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

185 **Acknowledgements**

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

## 193 AI Disclosure

194 Generative artificial intelligence was used during the development of this project in accordance  
 195 with the JOSS AI Usage Policy. Claude Sonnet 4.5 (Anthropic) was used to assist with code  
 196 generation, code refactoring, test scaffolding, and copy-editing of software-related materials  
 197 and documentation. Any AI-assisted outputs were reviewed, edited, and validated by the  
 198 human authors, who made all core design, architectural, and methodological decisions.

199 The manuscript itself was written entirely by hand by the authors and was not generated  
 200 or drafted using AI tools. The authors take full responsibility for the accuracy, originality,  
 201 licensing, and ethical and legal compliance of all submitted materials.

202 No generative AI tools were used for conversational interactions with editors or reviewers.

## 203 Appendix A: Task Formalization

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

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

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

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

208 This distinction shapes the training and inference stack. Multi-class models typically pair  
 209 a softmax activation with categorical cross-entropy, yielding normalized class probabilities  
 210 ([Goodfellow et al., 2016](#)). Multi-label classifiers instead apply independent sigmoid activations  
 211 with binary cross-entropy, producing class-wise confidence scores that require calibrated  
 212 thresholds at prediction time ([Goodfellow et al., 2016](#)). LabelFusion preserves these per-class  
 213 semantics when concatenating transformer logits and LLM scores, allowing the fusion network  
 214 to learn how much to trust each source under either regime.

## 215 References

- 216 Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of deep  
 217 bidirectional transformers for language understanding. *arXiv Preprint arXiv:1810.04805*.  
<https://doi.org/10.48550/arXiv.1810.04805>
- 218 Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.
- 219 Kant, G., Wiebelt, L., Weisser, C., Kis-Katos, K., Luber, M., & Säfken, B. (2022). An  
 220 iterative topic model filtering framework for short and noisy user-generated data: Analyzing  
 221 conspiracy theories on twitter. *International Journal of Data Science and Analytics*, 20(2),  
 222 269–289. <https://doi.org/10.1007/s41060-022-00321-4>
- 223 Kant, G., Zhelyazkov, I., Thielmann, A., Weisser, C., Schlee, M., Ehrling, C., Säfken, B., &  
 224 Kneib, T. (2024). One-way ticket to the moon? An NLP-based insight on the phenomenon  
 225 of small-scale neo-broker trading. *Social Network Analysis and Mining*, 14(1), 121. <https://doi.org/10.1007/s13278-024-01273-2>
- 226 Lewis, D. D. (1997). Reuters-21578 text categorization test collection, distribution 1.0. *KDD  
 227 Workshop on Text Mining*. <https://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html>
- 228 Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer,  
 229 L., & Stoyanov, V. (2019). RoBERTa: A robustly optimized BERT pretraining approach.  
 230 *arXiv Preprint arXiv:1907.11692*. <https://doi.org/10.48550/arXiv.1907.11692>

- 234 Luber, M., Weisser, C., Säfken, B., Silbersdorff, A., Kneib, T., & Kis-Katos, K. (2021).  
235 Identifying topical shifts in twitter streams: An integration of non-negative matrix  
236 factorisation, sentiment analysis and structural break models for large scale data. In  
237 J. Bright, A. Giachanou, V. Spaiser, F. Spezzano, A. George, & A. Pavliuc (Eds.),  
238 *Disinformation in open online media* (pp. 33–49). Springer International Publishing.  
239 [https://doi.org/10.1007/978-3-030-87031-7\\_3](https://doi.org/10.1007/978-3-030-87031-7_3)
- 240 OpenAI. (2023). GPT-4 technical report. *arXiv Preprint arXiv:2303.08774*. <https://doi.org/10.48550/arXiv.2303.08774>
- 241 Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin,  
242 Z., Gimelshein, N., Antiga, L., & others. (2019). PyTorch: An imperative style, high-  
243 performance deep learning library. *Advances in Neural Information Processing Systems*, 32.  
244 <https://doi.org/10.48550/arXiv.1912.01703>
- 245 Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel,  
246 M., Prettenhofer, P., Weiss, R., Dubourg, V., & others. (2011). Scikit-learn: Machine  
247 learning in python. *Journal of Machine Learning Research*, 12(Oct), 2825–2830. <http://www.jmlr.org/papers/v12/pedregosa11a.html>
- 248 Thielmann, A. F., Weisser, C., & Säfken, B. (2024). Human in the loop: How to effectively  
249 create coherent topics by manually labeling only a few documents per class. In N. Calzolari,  
250 M.-Y. Kan, V. Hoste, A. Lenci, S. Sakti, & N. Xue (Eds.), *Proceedings of the 2024 joint  
251 international conference on computational linguistics, language resources and evaluation  
252 (LREC-COLING 2024)* (pp. 8395–8405). ELRA; ICCL. <https://doi.org/10.48550/arXiv.2212.09422>
- 253 Thielmann, A., Weisser, C., & Krenz, A. (2021). One-class support vector machine and LDA  
254 topic model integration—evidence for AI patents. In N. H. Phuong & V. Kreinovich (Eds.),  
255 *Soft computing: Biomedical and related applications* (pp. 263–272). Springer International  
256 Publishing. [https://doi.org/10.1007/978-3-030-76620-7\\_23](https://doi.org/10.1007/978-3-030-76620-7_23)
- 257 Thielmann, A., Weisser, C., Krenz, A., & Säfken, B. (2021). Unsupervised document  
258 classification integrating web scraping, one-class SVM and LDA topic modelling. *Journal  
259 of Applied Statistics*, 50(3), 574–591. <https://doi.org/10.1080/02664763.2021.1919063>
- 260 Thormann, M.-L., Farchmin, J., Weisser, C., Kruse, R.-M., Säfken, B., & Silbersdorff, A.  
261 (2021). Stock price predictions with LSTM neural networks and twitter sentiment. *Statistics,  
262 Optimization & Information Computing*, 9(2), 268–287. <https://doi.org/10.19139/soic-2310-5070-1202>
- 263 Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T.,  
264 Louf, R., Funtowicz, M., & others. (2019). HuggingFace's transformers: State-of-the-art  
265 natural language processing. *arXiv Preprint arXiv:1910.03771*. <https://doi.org/10.48550/arXiv.1910.03771>
- 266 Zhang, X., Zhao, J., & LeCun, Y. (2015). Character-level convolutional networks for text  
267 classification. *Advances in Neural Information Processing Systems*, 28, 649–657. <https://doi.org/10.48550/arXiv.1509.01626>
- 268
- 269
- 270
- 271
- 272
- 273