Comparing Generated Pseudo-Labeled News Datasets: A Retrieval-and-Verification Approach

Giulio Alfarano[1,*,†], Stefano Scotta[1,*,†] and Alberto Messina[1]

[1]RAI - Centro Ricerche, Innovazione Tecnologica e Sperimentazione, Via Giovanni Carlo Cavalli 6, 10138, Turin, Italy

Abstract

We study the problem of constructing high-quality pseudo-labeled datasets for text classification when no labeled training data is available and the LLM annotation budget is limited. We compare direct LLM inference, LLM annotation of randomly sampled documents under multiple prompting regimes, unsupervised zero-shot scoring, and a retrieval-guided verification pipeline, within a general evaluation protocol that can be instantiated with a small human- or LLM-annotated validation set, even in the absence of a fully labeled corpus. Experiments on two public news benchmarks (single-label and multi-label) show that pseudo-labeled training is consistently preferable to direct LLM inference, and that our proposed retrieval-and-verification pipeline, which concentrates LLM queries on the most relevant candidates, is the strongest strategy across the considered settings. We further find that unsupervised methods equipped with carefully engineered category definitions are a surprisingly competitive and inexpensive alternative, particularly in single-label settings.

Keywords

Pseudo-labeling, LLM, Retrieval, News classification

1. Introduction

Text classification over large document collections is a fundamental task in natural language processing, with applications ranging from news categorization and content moderation to scientific and legal document analysis. In many practical scenarios, large corpora of unlabeled documents are readily available, whereas obtaining manually annotated training data is expensive or infeasible. The challenge is therefore to derive reliable label information from these collections without relying on large-scale manual annotation.

Recent large language models (LLMs) have demonstrated strong zero-shot and few-shot classification capabilities across a wide range of tasks (see [2, 3, 4]). A straightforward strategy consists of querying an LLM for every document requiring classification. While this approach eliminates the need for labeled training data, its cost scales linearly with corpus size. For proprietary models, this translates into recurring monetary expenses proportional to the number of processed documents. For open-weight models, the same strategy requires substantial computational infrastructure and sustained GPU memory occupancy during inference. Consequently, direct LLM-based classification becomes increasingly costly as document volume grows.

A common alternative is to use LLMs to generate pseudo-labels on a subset of documents and train a lightweight classifier on them (see [5], for example). This decouples annotation costs from deployment volume while retaining much of the predictive power of large language models. With a limited annotation budget, performance depends on the document selection strategy for labeling. In particular, uniform random sampling may waste a significant fraction of the available budget on irrelevant examples and can severely under represent minority categories in imbalanced corpora.

In this work, we study the problem of constructing high-quality pseudo-labeled datasets from large unlabeled corpora under a fixed LLM annotation budget, and we ask a more general question: among

NL4AI 2026: Ninth Workshop on Natural Language for Artificial Intelligence, October 6-9th, 2026, Perugia, Italy [1]

*Corresponding author.

†These authors contributed equally.

giulio.alfarano@rai.it (G. Alfarano); stefano.scotta@rai.it (S. Scotta); alberto.messina@rai.it (A. Messina)

0009-0001-9586-587X (G. Alfarano); 0000-0003-1078-2985 (S. Scotta); 0000-0002-8262-2449 (A. Messina)

© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

published 2026-11-05

CEUR-WS.org/Vol-4315/paper5.pdf

the strategies available for turning a limited number of LLM calls into a usable training set – random sampling under different prompting regimes, purely unsupervised zero-shot scoring, or retrieval-guided candidate selection – which one is preferable, and how should such strategies be compared in the first place? To this end, we introduce a general evaluation protocol for pseudo-labeling strategies under a fixed annotation budget, applicable whenever a small held-out labeled set is available for validation. Within this protocol we compare direct LLM inference, several random-sampling-plus-prompting baselines, two unsupervised zero-shot methods, and a retrieval-and-verification pipeline that ranks documents by lexical and semantic relevance to each target category before submitting only the most promising candidates to the LLM for verification.

We evaluate all strategies on two public single-label and multi-label news benchmarks. Across both settings, classifiers trained on pseudo-labeled data consistently outperform direct LLM inference on Macro F1 with a reduced annotation budget. Among the pseudo-labeling strategies, our retrieval-and-verification pipeline ranks among the strongest, although its advantage over the best prompting-based baselines is often modest. We additionally find that unsupervised zero-shot methods, when equipped with carefully engineered category definitions, are competitive – particularly in single-label settings – making them a reasonable low-cost alternative when even a modest annotation budget is unavailable.

The main contributions of this work are summarized as follows:

- We introduce a general evaluation protocol for comparing pseudo labeling strategies under a fixed annotation budget, instantiable with a small human- or LLM-annotated validation set even when no fully labeled corpus is available.
- We carry out a broad comparison of pseudo-labeling strategies – direct LLM inference, randomly sampled LLM annotation under multiple prompting regimes, unsupervised zero-shot scoring, and retrieval-guided verification across two public benchmarks.
- We present a retrieval-and-verification pipeline which consistently ranks among the best strategies compared.

1.1. Related Work

Weak supervision and pseudo-labeling. Learning from automatically generated labels has been extensively studied under the frameworks of distant and weak supervision. Distant supervision leverages external resources to heuristically assign labels to training instances [6], while programmatic weak supervision combines multiple noisy labeling functions through a generative model that estimates their reliability [7, 8]. More recently, large language models have emerged as effective annotators for generating pseudo-labels with minimal human intervention. Prior work has shown that LLM-generated annotations can achieve competitive quality across a variety of text classification tasks [3, 5]. Our approach follows the same objective of constructing training data automatically, but replaces hand-crafted labeling functions and corpus-wide annotation with retrieval-guided candidate selection followed by targeted LLM verification.

Retrieval-guided verification. Combining an inexpensive retrieval stage with a more computationally intensive verification or prediction stage is a common paradigm in information retrieval and open-domain question answering. Systems such as DrQA [9] and Dense Passage Retrieval [10] first identify promising candidates and subsequently apply more expensive downstream models. Similar retrieval-first strategies have also been adopted in extreme multi-label classification, where retrieval is used to narrow the search space before label prediction [11]. Our work adapts this principle to pseudo-label generation, using retrieval to prioritize documents that are most likely to yield positive annotations before invoking LLM-based verification.

Unsupervised document classification. Without labeled training data, text classification can be approached with unsupervised approaches, such as natural language inference (NLI) or embedding similarity. The NLI paradigm, introduced in [4], reformulates label assignment as binary entailment

between a document and a hypothesis derived from each candidate label, exploiting the generalization of cross-encoders trained on large entailment corpora. A complementary approach involves text embedding: a transformer-based encoder is used to codify both documents and label descriptions into a shared dense space and assigns labels via cosine similarity; Sentence-BERT [12] established the foundations of this paradigm, later extended by instruction-tuned embedding models that improve discriminability through asymmetric task-specific prompting [13]. A recent systematic comparison across both families [14] shows that NLI cross-encoders plateau as backbone size increases, while strong embedding models offer a competitive accuracy-latency trade-off. We adopt both approaches as unsupervised baselines: a DeBERTa-based NLI cross-encoder [15] and a Qwen3-Embedding-0.6B model with artificially generated label definitions. This allows us to evaluate whether either provides sufficient signal to bootstrap a downstream supervised classifier without any manually annotated data.

2. Problem Setting and Methods

Let $\mathcal{D} = \mathcal{D}_{\text{train}} \cup \mathcal{D}_{\text{test}}$ be a corpus of textual documents and $\mathcal{Y} = \{c_1, \ldots, c_m\}$ a set of labels. Each text $x \in \mathcal{D}$ is associated with a set of labels $\{y_x^i\}_{i=1}^{n_x}$, where $y_x^i \in \mathcal{Y}$ and $0 \leq n_x \leq m$ denotes the number of labels assigned to x. This includes mono and multilabel classification problems.

The standard classification problem consists of defining a mapping that assigns to each $x \in \mathcal{D}_{\text{test}}$ a set of predicted labels $\{\hat{y}_x^i\}_{i=1}^{n_x} \subseteq \mathcal{Y}$, minimizing the discrepancy with respect to the corresponding ground-truth label set $\{y_x^i\}_{i=1}^{n_x}$. The test set $\mathcal{D}_{\text{test}}$ is therefore used to evaluate the performance of the learned model, while the ultimate goal is to generalize to previously unseen and unlabeled documents drawn from the same distribution.

2.1. Classification with Labeled Training Data

When a labeled training dataset $\mathcal{D}_{\text{train}}$ is available, a parametric model $f_\theta : \mathcal{X} \to 2^{\mathcal{Y}}$ can be learned via standard supervised learning: given $\mathcal{D}_{\text{train}} = \{(x_j, \{y_{x_j}^i\}_{i=1}^{n_{x_j}})\}_{j=1}^{|\mathcal{D}_{\text{train}}|}$, the optimal parameters θ^* are obtained by minimizing a suitable loss between predicted and ground-truth label sets. The resulting model f_{θ^*} assigns to each $x \in \mathcal{D}_{\text{test}}$ a predicted label set $\{\hat{y}_x^i\}_{i=1}^{\hat{n}_x} = f_{\theta^*}(x)$. We use this fully supervised setting as the upper-bound reference throughout our evaluation protocol.

2.2. Classification without Labeled Training Data

When no labeled training set is available, the problem becomes substantially more challenging. In this setting, one must rely on alternative sources of supervision, such as direct LLM inference, automatically generated pseudo-labels, or fully unsupervised structure discovery. We briefly review these strategies before introducing our method.

2.2.1. Direct LLM-based Classification

With the advent of large language models (LLMs), a straightforward strategy is to use them to directly assign labels to each $x \in \mathcal{D}_{\text{test}}$. In practice, this is typically achieved by prompting the model to return the set of categories associated with a given document. Various prompting strategies can be employed, including zero-shot, few-shot and definition-enriched prompting (see Section 3.4.1 for details). This approach is effective but does not scale due to inference costs. Moreover, since the test set serves only as a proxy for real-world deployment scenarios, applying this method at scale to large collections of unseen and unlabeled documents further amplifies these costs.

2.2.2. Unsupervised methods

An alternative approach to labeling $\mathcal{D}_{\text{test}}$ relies on unsupervised (i.e., requiring no labeled data from the target dataset, though the underlying models were pretrained on large text corpora) methods that

classify samples from features extracted directly from the text. We selected the two most suitable approaches of this kind for the current problem.

Natural language inference NLI aims to determine the directional logical relation between a *premise* p and a *hypothesis* h: a model f predicts $r \in \{\textit{entailment}, \textit{neutral}, \textit{contradiction}\}$, where entailment holds if h follows from p, contradiction if h opposes p, and neutral otherwise. The *zero-shot trick* [4] repurposes this for zero-shot multi-label classification: given document x and candidate category $c_k \in \mathcal{Y}$, a hypothesis *"This text is about c_k"* is paired with x as premise, and the entailment probability serves as relevance score $s_k(x) \in (0, 1)$. In the *multi-label* setting, each c_k is scored independently and x is assigned to c_k whenever $s_k(x) > \tau_k$, yielding $\hat{\mathcal{Y}}(x) \subseteq \mathcal{Y}$; in the *mono-label* setting, scores for all $|\mathcal{Y}|$ categories are computed jointly and $\hat{y}(x) = \arg\max_k \; s_k(x)$, making τ_k irrelevant. Either way, total inference cost is $|\mathcal{Y}| \cdot |\mathcal{D}_{\text{test}}|$ forward passes, since each document-label pair needs a separate cross-encoder pass. No labeled data is required, but classification quality depends entirely on how well the hypothesis templates capture each category's semantics.

Text Embedding Similarity Dense embedding models map text to fixed-dimensional vectors such that semantically related texts are geometrically close. Given encoder $\phi : \mathcal{X} \to \mathbb{R}^d$, we adopt an asymmetric scheme: each document x is encoded with a retrieval prompt q as $\phi_q(x)$, while each category $c_k \in \mathcal{Y}$ is represented by a generated textual definition d_k (as in 3.2), encoded as $\phi(d_k)$. All vectors are L2-normalised, so the relevance score is the dot product $s_k(x) = \phi_q(x)^\top \phi(d_k) \in (-1, 1)$. As with NLI, x is assigned via $s_k(x) > \tau_k$ (multi-label) or $\hat{y}(x) = \arg\max_k \; s_k(x)$ (mono-label, threshold-free). Since label embeddings $\{\phi(d_k)\}_{k=1}^{|\mathcal{Y}|}$ are computed once and reused, total inference reduces to $|\mathcal{D}_{\text{test}}| + |\mathcal{Y}|$ passes. No labeled data is required; the main limitation is sensitivity to the quality of $\{d_k\}$ and, in the multi-label setting, to τ_k, since cosine similarity is less naturally calibrated around a fixed boundary.

2.2.3. LLM Annotation with Downstream Classifier

An intermediate strategy consists of using the LLM exclusively to annotate a subset of the corpus, and subsequently training a parametric classifier f_θ on the resulting pseudo-labeled dataset. Suppose one wishes to form a dataset $\widehat{\mathcal{D}}_{\text{train}} \subseteq \mathcal{D}_{\text{train}}$ of some arbitrary size. Since labels are unknown, documents are sampled uniformly from the corpus. To classify them the LLM is queried to produce, for each inspected document $x \in \widehat{\mathcal{D}}_{\text{train}}$, a predicted label set $\{\hat{y}_x^i\}_{i=1}^{n_x}$, using, for example, the same prompting techniques mentioned in Section 2.2.1. Once the classifier f_{θ^*} has been trained on $\widehat{\mathcal{D}}_{\text{train}}$, no further LLM queries are needed at inference time. Performance depends entirely on the quality of the pseudo-labels.

2.3. Proposed Method: Retrieval-and-Verification Pipeline

The proposed approach constructs a pseudo-labeled training set $\widetilde{\mathcal{D}}_{\text{train}} \subseteq \mathcal{D}_{\text{train}}$ through a retrieval-and-verification pipeline, rather than by querying the LLM indiscriminately on sampled articles. The key idea is to use inexpensive retrieval signals to produce a single unified ranking of the entire corpus, and to submit to the LLM only a selected subset of documents.

Corpus preprocessing. Before retrieval, each document $x \in \mathcal{D}_{\text{train}}$ undergoes a preprocessing pipeline that produces the representations required by the retrieval modules described below. First, each document is associated with a set of semantic tags $\mathcal{T}_x$, for instance by means of an automated tagging procedure. Second, a TF–IDF vectorizer is fitted on a sufficiently large subset $\mathcal{D}_{\text{tfidf}} \subseteq \mathcal{D}$ and is then used to compute, for each $x \in \mathcal{D}$, a TF–IDF representation $\phi_{\text{tfidf}}(x)$. Third, a dense semantic embedding $\phi_{\text{emb}}(x)$ is computed for each document using a sentence-transformer model. All representations are computed offline once and stored before retrieval.

Phase 1 — Generation of operational category definitions. For each category $c_k \in \mathcal{Y}$, a single LLM call is issued to generate an operational definition of the category, structured as a triple $(d_k, \mathcal{T}_k, \mathcal{P}_k)$, where d_k is a natural language definition of c_k (including boundary cases to be excluded), $\mathcal{T}_k$ is a set of n_t short tags representative of the category, and $\mathcal{P}_k$ is a set of n_p semantically representative phrases in headline style, suited for lexical and semantic retrieval. The prompt includes the full list of categories $\mathcal{Y}$, so as to encourage the LLM to minimize lexical and semantic overlap across the generated representations. This first phase requires exactly $|\mathcal{Y}|$ LLM calls, independently of the corpus size.

Remark 1. *A preliminary category-definition step of this kind could also be incorporated into the direct LLM baselines of Sections 2.2.1 and 2.2.3, since more precise and operational label descriptions may improve labeling consistency.*

Phase 2 — Construction of a unified ranking. For each category $c_k \in \mathcal{Y}$ and each document $x \in \mathcal{D}$, three complementary retrieval components are computed independently and then min–max normalized over the corpus. The first is a tag-overlap score

$$s_{\text{tag}}^{\alpha,\beta}(x, c_k) = \frac{|\mathcal{T}_x \cap \mathcal{T}_k|^\alpha}{|\mathcal{T}_x \cup \mathcal{T}_k|^\beta}, \tag{1}$$

for some positive $\alpha, \beta \in \mathbb{R}$, where $\mathcal{T}_x$ denotes the set of tags associated with document x after lowercasing and stopword removal. The second is a TF–IDF score $s_{\text{tfidf}}(x, c_k)$, defined as the maximum cosine similarity between $\phi_{\text{tfidf}}(x)$ and the TF–IDF vectors of the phrases in $\mathcal{P}_k$. The third is a dense embedding score $s_{\text{emb}}(x, c_k)$, defined as the maximum cosine similarity between $\phi_{\text{emb}}(x)$ and the embeddings of the phrases in $\mathcal{P}_k$. After min–max normalization, the three components are summed to yield a per-category retrieval score

$$s(x, c_k) = \bar{s}_{\text{tag}}^{\alpha,\beta}(x, c_k) + \bar{s}_{\text{tfidf}}(x, c_k) + \bar{s}_{\text{emb}}(x, c_k) \ \in [0,\ 3]. \tag{2}$$

The overall relevance of document x to the entire label set $\mathcal{Y}$ is then captured by the unified score

$$S(x) = \sum_{k=1}^{|\mathcal{Y}|} s(x, c_k), \tag{3}$$

which aggregates, across all categories, the degree to which x matches the lexical and semantic signatures of $\mathcal{Y}$. The corpus is sorted in decreasing order of $S(x)$, yielding a single ranked list in which documents broadly relevant to the categories appear near the top, while documents with low affinity to all categories are pushed to the bottom.

Phase 3 — LLM verification: top-N and bottom-M passes. The unified ranking is used to select two disjoint subsets of documents for LLM verification.

Top-N pass. The N highest-scoring documents are submitted to the LLM for classification (multi or mono label depending on the type of dataset that we want to build). Each document x is presented together with the full list of categories $\mathcal{Y}$ and all operational definitions $\{d_k\}$. The LLM is instructed to return the subset of categories in $\mathcal{Y}$ that apply to x—possibly one, more than one, or none. In this part, we define whether the dataset is single-label or multi-label and whether a "none" category is included to represent instances that do not belong to any category in $\mathcal{Y}$. Documents are assigned all categories for which the LLM returns a positive response and are included in $\widetilde{\mathcal{D}}_{\text{train}}$ with the corresponding annotation.

Bottom-M pass (optional). In the case in which the dataset that we want to build admits documents not associated to any categories we can select negative examples as follows. The M lowest-scoring documents are submitted to the LLM for binary negative confirmation. Each document is presented alongside the full category list and definitions, and the model is instructed to answer `true` if the document belongs to none of the categories (confirming it as a true negative), or `false` if it matches at least one (signaling a retrieval miss). Documents confirmed as true negatives are included in $\widetilde{\mathcal{D}}_{\text{train}}$ as shared negative examples across all categories.

Training of the classifier. Once $\widetilde{\mathcal{D}}_{\text{train}}$ has been constructed, a parametric classifier f_θ is trained on it as described in Section 2.2.3. The learned model f_{θ^*} can then be used to assign labels to any document $x \in \mathcal{D}_{\text{test}}$ without requiring further LLM queries.

2.4. Evaluation Framework

To assess the effectiveness of the proposed approach, we compare the following families of methods on the held-out set $\mathcal{D}_{\text{test}}$, using standard classification metrics:

1. **Supervised Upper Bound:** A parametric model f_{θ^*} trained on ground-truth labels (Section 2.1).
2. **Direct LLM Classifiers:** Zero-shot labeling via direct LLM querying (Section 2.2.1).
3. **Unsupervised methods:** Zero-shot classification via NLI entailment scoring and text embedding similarity, requiring no labeled data beyond the category schema (Section 2.2.2).
4. **Pseudo-labeled Classifiers:** A parametric model f_{θ^*} trained on labels generated either via unguided LLM annotation (Section 2.2.3) or through our Retrieval-and-Verification pipeline (Section 2.3).

To ensure a fair comparison among pseudo-labeling strategies, all resulting pseudo-labeled datasets contain exactly the same number of training samples, and all downstream parametric classifiers are trained using the same architecture and optimization procedure. Therefore, differences in predictive performance can be attributed to the quality and selection of the generated labels, rather than to differences in training-set size or classifier capacity. We regard the method as successful if it outperforms the classifier trained on the dataset obtained through straightforward LLM annotation of a random subset of the training set (Section 2.2.3). Such a result would indicate that the proposed Retrieval-and-Verification pipeline generates higher-quality pseudo-labels than direct LLM annotation under the same annotation budget. Furthermore, if the resulting classifier achieves performance comparable to the Direct LLM classifier (Section 2.2.1), it demonstrates that similar predictive accuracy can be obtained while replacing repeated inference-time LLM calls with a single trained classifier, substantially reducing computational cost. Finally, performance close to the supervised upper bound would indicate that the automatically generated labels approach the quality of human annotations.

3. Experimental Setup

To evaluate the proposed retrieval-and-verification framework, we conduct experiments on two publicly available news datasets representing complementary classification settings: a single-label benchmark and a multi-label benchmark. Unless otherwise specified, all experimental comparisons are performed using identical downstream training procedures, so that differences in performance can be attributed exclusively to the quality of the generated labels.

We evaluate the pipeline by training a downstream classifier on each pseudo-labeled dataset; Section 3.5 describes the training setup and details, while Section 4 compares the resulting classifiers against each other, against the unsupervised methods introduced in Section 2.2.2, and against direct LLM classification of the test sets.

3.1. Datasets and Splits

3.1.1. Single-Label Setting: MN-DS

The single-label experiments are conducted on the *MN-DS* dataset of labeled news articles[1] ([16]). Since the dataset is organized hierarchically, we restrict our analysis to the 17 first-level categories. The corpus contains a total of $10,917$ documents. We randomly partition it into a training set $\mathcal{D}^{\text{SL}}_{\text{train}}$ containing $8,733$ articles and a held-out test set $\mathcal{D}^{\text{SL}}_{\text{test}}$ containing $2,184$ articles. Each document is

[1] https://zenodo.org/records/7394851

associated with exactly one category, excluding both multi-label assignments and unlabeled examples. The empirical class distribution of the resulting split is reported in Table 1. As expected, the dataset exhibits a noticeable class imbalance.

Table 1
Category distribution for the single-label MN-DS dataset.

Category	$\mathcal{D}^{\mathrm{SL}}_{\mathrm{train}}$ (%)	$\mathcal{D}^{\mathrm{SL}}_{\mathrm{test}}$(%)	Category	$\mathcal{D}^{\mathrm{SL}}_{\mathrm{train}}$(%)	$\mathcal{D}^{\mathrm{SL}}_{\mathrm{test}}$(%)
Society	895 (10.2)	205 (9.4)	Environment	475 (5.4)	125 (5.7)
Sport	725 (8.3)	182 (8.3)	Disaster/Accident/Emerg.	407 (4.7)	93 (4.3)
Politics	722 (8.3)	178 (8.2)	Crime, Law and Justice	399 (4.6)	101 (4.6)
Science and Technology	648 (7.4)	152 (7.0)	Economy/Business/Finance	316 (3.6)	84 (3.8)
Conflict, War and Peace	639 (7.3)	161 (7.4)	Weather	314 (3.6)	86 (3.9)
Religion and Belief	621 (7.1)	179 (8.2)	Lifestyle and Leisure	235 (2.7)	65 (3.0)
Health	564 (6.5)	136 (6.2)	Arts, Culture, Ent. & Media	232 (2.7)	68 (3.1)
Labour	563 (6.4)	140 (6.4)			
Education	496 (5.7)	111 (5.1)	**Total**	**8,733**	**2,184**
Human Interest	482 (5.5)	118 (5.4)			

3.1.2. Multi-Label Setting: Newswire

The multi-label experiments are conducted on a subset of the *dell-research-harvard/newswire* corpus[2] ([17]). Following the temporal split provided by the dataset, the training set $\mathcal{D}^{\mathrm{ML}}_{\mathrm{train}}$ contains all $188,334$ articles published between 1970 and 1976, whereas the held-out test set $\mathcal{D}^{\mathrm{ML}}_{\mathrm{test}}$ contains the $21,029$ articles published in 1977. This temporal partition allows us to evaluate the generalization capability of the downstream classifiers on previously unseen time periods. Unlike MN-DS, documents in this corpus may belong to multiple categories or may remain completely unlabeled. Table 2 reports the empirical label distribution for both splits. The category frequencies remain largely stable across the temporal partition, with the exception of the unlabeled documents.

Table 2
Category distribution for the multi-label Newswire dataset.

Category	$\mathcal{D}^{\mathrm{ML}}_{\mathrm{train}}$(%)	$\mathcal{D}^{\mathrm{ML}}_{\mathrm{test}}$(%)
Antitrust	126 (0.07)	12 (0.06)
Civil Rights	4,844 (2.57)	347 (1.65)
Crime	24,673 (13.10)	2,818 (13.40)
Govt. Regulation	25,972 (13.79)	3,670 (17.45)
Labor Movement	13,228 (7.02)	1,386 (6.59)
Politics	76,554 (40.65)	8,839 (42.03)
Protests	2,886 (1.53)	244 (1.16)
No Category	74,330 (39.47)	3,094 (14.71)

3.2. Operational Category Definitions

As described in Section 2.3, the retrieval-and-verification pipeline requires an operational representation (d_k, T_k, P_k) for every category $c_k \in \mathcal{Y}$. These descriptors were generated offline using `gpt-5.1` ([18]) through a standardized prompt applied uniformly to all categories in both datasets. We generated $n_t = 50$ tags and $n_p = 30$ representative phrases per category.

3.3. Retrieval-and-Verification Dataset Construction

The pseudo-labeled training sets were constructed by executing the retrieval-and-verification pipeline described in Section 2.3. Unless otherwise specified, the following implementation choices are shared

[2] https://huggingface.co/datasets/dell-research-harvard/newswire

across both datasets.

Retrieval Preprocessing. For each training corpus $\mathcal{D}^{\ell}_{\text{train}}$, $\ell \in \{\text{SL}, \text{ML}\}$, three complementary representations are computed offline. Editorial tags $\mathcal{T}_x$ are assigned to each document using `raicrits/Llama_NewsTagger_XRECO`[3], a LLaMA-based model fine-tuned for news tagging. A TF–IDF vectorizer is fitted exclusively on the corresponding training set, constructing the sparse retrieval representation directly from the available corpus. Dense embeddings $\phi_{\text{emb}}(x)$ are produced by `Qwen/Qwen3-Embedding-0.6B` [13][4]. The three per-category components are independently min–max normalized and combined via (2) with $\alpha = \beta = 1$ (i.e., standard Intersection-over-Union for the tag-overlap score), yielding the unified ranking score $S(x)$ as in (3).

Verification Step. The downstream verification strategy differs according to the characteristics of the target dataset.

- **Single-Label dataset.** For the MN-DS dataset, every document is guaranteed to belong to exactly one category. Consequently, the negative confirmation stage is unnecessary and the verification process is restricted to the top-N_{single} highest-ranked documents only, with N fixed to $2,994$. So, after the verification process, we obtain the dataset $\widetilde{\mathcal{D}}^{\text{SL}}_{\text{train}}$ containing exactly $2,994$ articles labeled with a single label each.
- **Multi-Label dataset.** For the Newswire dataset, LLM verification was performed on the $N = 10,000$ highest-ranked documents (top-N verification pass). Since a large fraction of the corpus contains documents that do not belong to any category, we additionally executed a negative confirmation pass on the M articles with the lowest score —as explained in Section 2.3 (Phase 3)— with M set to $\lfloor N/|\mathcal{Y}| \rfloor = \lfloor 10,000/7 \rfloor = 1,428$. Of the $1,428$ candidates, 91 were rejected as false negatives, yielding $1,337$ confirmed negatives. The resulting dataset $\widetilde{\mathcal{D}}^{\text{ML}}_{\text{train}}$ is therefore composed of $11,337$ labeled articles of which $1,337$ negative examples coming from the bottom M verification steps.

3.4. Random Annotation Baselines

To assess the contribution of the proposed retrieval stage, we compare our approach against pseudo-labeled datasets obtained through direct LLM annotation of uniformly sampled training documents. For each setting, we randomly sample documents from the corresponding training set, matching the size of the dataset built with our Retrieval-and-Verification method, so that the two pseudo-labeled datasets — and the subsequent downstream training on them — remain directly comparable.

- **Single-Label dataset.** For the MN-DS dataset, we uniformly sample a subset $\widehat{\mathcal{D}}^{\text{SL}}_{\text{train}} \subset \mathcal{D}^{\text{SL}}_{\text{train}}$ containing exactly $2,994$ documents, matching the size of $\widetilde{\mathcal{D}}^{\text{SL}}_{\text{train}}$. Each sampled document is then annotated by direct LLM labeling, without any retrieval or verification step.
- **Multi-Label dataset.** For the Newswire dataset, we uniformly sample a subset $\widehat{\mathcal{D}}^{\text{ML}}_{\text{train}} \subset \mathcal{D}^{\text{ML}}_{\text{train}}$ containing exactly $11,337$ documents, matching the size of $\widetilde{\mathcal{D}}^{\text{ML}}_{\text{train}}$. As in the single-label case, each $x \in \widehat{\mathcal{D}}^{\text{ML}}_{\text{train}}$ is directly annotated by the LLM, bypassing the retrieval-and-verification pipeline.

3.4.1. Annotation Techniques and Reference Labels

For $\ell \in \{\text{SL}, \text{ML}\}$, the documents in $\widehat{\mathcal{D}}^{\ell}_{\text{train}}$ are annotated independently using `Qwen/Qwen3-VL-32B-Instruct` ([19]) under three, prompting configurations (depending on ℓ):

- **Zero-shot.** The prompt contains only the category names.

[3] https://huggingface.co/raicrits/Llama_NewsTagger_XRECO
[4] https://huggingface.co/Qwen/Qwen3-Embedding-0.6B

- **Few-shot.** The prompt additionally includes manually selected examples for each category.
- **Definition-augmented.** The prompt incorporates the operational category definitions d_k generated offline with `gpt-5.1`.

This procedure produces respectively, for both the single-label and the multi-label cases, the pseudo-labeled datasets $\widehat{\mathcal{D}}^{\ell}_{\mathrm{ZS}}$, $\widehat{\mathcal{D}}^{\ell}_{\mathrm{FS}}$, $\widehat{\mathcal{D}}^{\ell}_{\mathrm{DEF}}$, for $\ell \in \{\mathrm{SL}, \mathrm{ML}\}$.

The ground-truth annotations associated with the documents in $\widehat{\mathcal{D}}^{\mathrm{SL}}_{\mathrm{train}}$ and $\widehat{\mathcal{D}}^{\mathrm{ML}}_{\mathrm{train}}$ are retained to construct the corresponding supervised reference datasets $\widehat{\mathcal{D}}^{\mathrm{SL}}_{\mathrm{GT}}$ and $\widehat{\mathcal{D}}^{\mathrm{ML}}_{\mathrm{GT}}$, each containing the same $2,994$ and $11,337$ documents respectively, but with their original gold labels.

3.5. Downstream classifiers

3.5.1. Classifier

To evaluate the quality of each pseudo-labeled dataset, for any $\ell \in \{\mathrm{SL}, \mathrm{ML}\}$, we train parametric classifiers f_{θ^*} on $\overline{\mathcal{D}}_{\mathrm{train}} \in \{\widetilde{\mathcal{D}}^{\ell}_{\mathrm{train}}, \widehat{\mathcal{D}}^{\ell}_{\mathrm{ZS}}, \widehat{\mathcal{D}}^{\ell}_{\mathrm{FS}}, \widehat{\mathcal{D}}^{\ell}_{\mathrm{DEF}}, \widehat{\mathcal{D}}^{\ell}_{\mathrm{GT}}\}$, for $\ell \in \{\mathrm{SL}, \mathrm{ML}\}$. To prevent any leakage onto the fixed benchmark, $\overline{\mathcal{D}}_{\mathrm{train}}$ is partitioned into two stratified subsets at an 80:20 ratio, denoted $\overline{\mathcal{D}}_{\mathrm{train}^*}$ and $\overline{\mathcal{D}}_{\mathrm{dev}}$, with seed 42 for reproducibility. The model is evaluated on $\overline{\mathcal{D}}_{\mathrm{dev}}$ after every epoch to drive both model and per-label threshold selection, leaving $\mathcal{D}^{\ell}_{\mathrm{test}}$ entirely untouched. All reported metrics are computed on $\mathcal{D}^{\ell}_{\mathrm{test}}$ exclusively.

We adopt the same architecture across all experimental conditions: RoBERTa-base [20], a state-of-the-art transformer encoder for text classification. The output layer is adapted to the classification objective. In the *multi-label* setting, each logit is passed through an independent sigmoid, producing a per-label probability in $(0, 1)$, and the model is trained with binary cross-entropy loss computed independently per label, so that multiple categories can be simultaneously assigned to a single document. For evaluation, we report results under a threshold regime: the per-label thresholds $\{\tau^*_k\}_{k=1}^{|\mathcal{Y}|}$ are selected greedily on $\overline{\mathcal{D}}_{\mathrm{dev}}$ to maximise per-label F1. In the *mono-label* setting, the logits are passed through a softmax, producing a probability distribution over $\mathcal{Y}$ that sums to one; the predicted label is $\hat{y}(x) = \arg\max_k \ f_{\theta^*}(x)_k$, and the loss is standard cross-entropy. To ensure comparability across all conditions, all models share the same hyperparameters: 3 epochs, learning rate 10^{-5}, AdamW optimizer and batch size 8.

3.5.2. Unsupervised approaches

Both unsupervised baselines operate exclusively on test sets: no labeled data is consulted at any stage, and classification relies solely on the semantic content of each document and the category schema. In the *mono-label* setting, both methods reduce to $\hat{y}(x) = \arg\max_k \ s_k(x)$, making threshold selection irrelevant. In the *multi-label* setting, a fixed threshold must be chosen without access to any labeled validation set. We set $\tau = 0.5$ for NLI, as this corresponds to the natural decision boundary of the entailment probability; for text embedding similarity we set $\tau = 0.3$, since cosine similarity scores are not calibrated in the same range and a threshold of 0.5 would cause most predictions to collapse to the *none* class (no label is selected). Unlike the trained classifiers, whose per-label thresholds are tuned on $\mathcal{D}_{dev}$, these thresholds were not tuned on any held-out data, making the competitive performance of the unsupervised baselines a conservative estimate.

4. Results

Throughout this section, we use the notation introduced in Section 3.4.1 to denote the training datasets. Table 3 and Table 4 report classification performance on $\mathcal{D}^{\mathrm{SL}}_{\mathrm{test}}$ and $\mathcal{D}^{\mathrm{ML}}_{\mathrm{test}}$, respectively, comparing the three method families introduced in Section 2.3: *Trained Classifiers* (Section 3.5), *Direct LLM Classifiers* (Section 2.2.1, obtained by prompting `Qwen/Qwen3-VL-32B-Instruct` on each test document with the same strategies presented in Section 3.4.1, with no classifier training involved), and *Unsupervised Classifiers* (Section 2.2.2), which require neither labeled data nor LLM queries (except for the categories definitions).

Method	Accuracy	Macro P	Macro R	Macro F1	Weighted F1
Trained Classifiers					
$\widehat{\mathcal{D}}^{\mathrm{SL}}_{\mathrm{GT}}$	*0.773*	*0.775*	*0.762*	*0.761*	*0.767*
$\widetilde{\mathcal{D}}^{\mathrm{SL}}_{\mathrm{train}}$ (Ours)	**0.590**	**0.591**	**0.608**	**0.572**	0.572
$\widehat{\mathcal{D}}^{\mathrm{SL}}_{\mathrm{ZS}}$	0.557	0.561	0.554	0.508	0.533
$\widehat{\mathcal{D}}^{\mathrm{SL}}_{\mathrm{FS}}$	0.584	0.561	0.577	0.532	0.558
$\widehat{\mathcal{D}}^{\mathrm{SL}}_{\mathrm{DEF}}$	0.566	0.551	0.563	0.518	0.547
Direct LLM Classification					
Zero-shot	0.552	0.478	0.469	0.441	0.540
Few-shot	0.552	0.396	0.394	0.370	0.542
Definition-augmented	0.583	0.482	0.484	0.467	**0.577**
Unsupervised Classifiers					
NLI (zero-shot trick)	0.319	0.428	0.317	0.323	0.332
Text Embedding Similarity	0.585	0.562	0.597	0.557	0.558

Table 3
Results on the single-label test set $\mathcal{D}^{\mathrm{SL}}_{\mathrm{test}}$. Training datasets are reported for the *Trained Classifiers*. Bold values indicate the top performance across the approaches, excluding the classifier trained on $\widehat{\mathcal{D}}^{\mathrm{SL}}_{\mathrm{GT}}$ which serves as upper-bound.

	Micro			Macro			Weighted			
Method	P	R	F1	P	R	F1	P	R	F1	EM
Trained Classifiers										
$\widehat{\mathcal{D}}^{\mathrm{ML}}_{\mathrm{GT}}$	0.918	0.892	0.905	0.743	0.728	0.733	0.919	0.892	0.904	0.850
$\widetilde{\mathcal{D}}^{\mathrm{ML}}_{\mathrm{train}}$ (Ours)	**0.647**	0.663	**0.655**	**0.495**	0.617	**0.499**	**0.737**	0.663	**0.673**	**0.529**
$\widehat{\mathcal{D}}^{\mathrm{ML}}_{\mathrm{ZS}}$	0.449	**0.750**	0.562	0.399	**0.740**	0.425	0.677	**0.750**	0.617	0.280
$\widehat{\mathcal{D}}^{\mathrm{ML}}_{\mathrm{FS}}$	0.602	0.720	**0.655**	0.474	0.679	0.490	**0.737**	0.720	0.666	0.483
$\widehat{\mathcal{D}}^{\mathrm{ML}}_{\mathrm{DEF}}$	0.645	0.662	0.654	0.482	0.631	0.497	0.731	0.662	0.672	0.525
Direct LLM Classification										
Zero-shot	0.458	0.742	0.566	0.390	0.711	0.422	0.664	0.742	0.617	0.292
Few-shot	0.592	0.714	0.647	0.463	0.673	0.477	0.723	0.714	0.652	0.476
Definition-augmented	0.641	0.663	0.652	0.481	0.619	0.492	0.722	0.663	0.672	0.523
Unsupervised Classifiers										
NLI (zero-shot trick)	0.530	0.609	0.567	0.354	0.467	0.386	0.601	0.609	0.596	0.461
Text Embedding Similarity	0.460	0.554	0.503	0.458	0.608	0.471	0.701	0.554	0.575	0.369

Table 4
Results on the multi-label test set $\mathcal{D}^{\mathrm{ML}}_{\mathrm{test}}$. Training datasets are reported for the *Trained Classifiers*. Bold values indicate the top performance across the approaches, excluding the classifier trained on $\widehat{\mathcal{D}}^{\mathrm{ML}}_{\mathrm{GT}}$ which serves as upper-bound.

The two tables report different metric subsets because the settings are not equivalent. Macro metrics average per-category Precision (P)/Recall (R)/F1 uniformly, Weighted metrics weight by category support, and Micro metrics pool true/false positives/negatives across categories before computing a single P/R/F1 — making Macro F1 most sensitive to performance on infrequent categories. In the single-label case, every classifier predicts exactly one category, so Micro P/R/F1 would coincide with Accuracy; we therefore report only Accuracy, Weighted F1, and Macro P/R/F1. In the multi-label case, predicted and gold label sets can differ in size, so Micro, Macro, and Weighted P/R/F1 are all independently informative and reported in full, alongside Exact Match (EM), the fraction of documents whose predicted label set exactly matches the gold set.

Beyond the fully supervised upper bound, the clearest pattern across both benchmarks is that training a lightweight classifier on pseudo-labeled data consistently beats querying an LLM directly at inference time on Macro F1, regardless of prompting strategy or how training documents were selected and labeled — confirming that offline computation during dataset construction is a better use of a fixed budget than repeated inference-time LLM calls.

Our retrieval-and-verification pipeline ($\tilde{\mathcal{D}}^{SL}_{\text{train}}$) achieves the best single-label Accuracy (0.590), though by a tiny margin over the strongest prompting baseline (few-shot, 0.584). The gap is larger in Macro F1 — more sensitive to the class imbalance in Table 1 — where our pipeline reaches 0.572, four points above the second best and five to six points above the rest of the classifiers trained on pseudo-labeled datasets, indicating the gain is concentrated in minority-class treatment rather than overall accuracy.

A more notable finding (consistent with what found in [14]) is how competitive unsupervised baselines are here: Text Embedding Similarity, requiring neither labeled data nor a single LLM call, reaches 0.585 Accuracy and 0.557 Macro F1 — essentially matching several pseudo-labeled classifiers and trailing our pipeline by half a point of Accuracy. With carefully engineered category definitions, a purely unsupervised embedding-based classifier can be thus considered as a reference baseline rather than a weak lower bound, and an attractive option when even a modest LLM annotation budget is unavailable.

On the multi-label benchmark (Table 4), all classifiers trained on pseudo-labeled datasets achieve remarkably similar performance, suggesting that downstream classifier training is generally robust to the specific pseudo-labeling strategy. Within this group, our method ($\widetilde{\mathcal{D}}^{\text{ML}}_{\text{train}}$) consistently ranks among the strongest approaches, matching the best Micro F1 while achieving the highest Macro F1 (0.499) and Weighted F1 (0.673). These results indicate that our retrieval-and-verification pipeline produces pseudo-labels that better preserve minority classes, leading to more balanced predictions without sacrificing overall accuracy. Unlike the single-label setting, unsupervised approaches degrade substantially, suggesting that more complex annotation tasks benefit from pseudo-labeled training data.

4.1. Label Assignment Quality

The evaluation in Section 4 measures the *downstream* effect of pseudo-labeling — how well a classifier trained on each pseudo-labeled dataset generalizes to the held-out test set. A complementary, more direct question is how faithfully the labels assigned during dataset construction match the true category of each document. Since MN-DS and the training portion of Newswire are themselves fully human-annotated, documents in $\widehat{\mathcal{D}}^{\ell}_{\text{ZS}}$, $\widehat{\mathcal{D}}^{\ell}_{\text{FS}}$, $\widehat{\mathcal{D}}^{\ell}_{\text{DEF}}$, and $\widetilde{\mathcal{D}}^{\ell}_{\text{train}}$ can be paired with the gold labels retained for the articles in them (for $\ell \in \{\text{SL}, \text{ML}\}$), letting us score the pseudo-labels directly. This analysis is only possible in our controlled benchmark setting, where gold labels are available, and serves solely as an internal validity check. All four methods were scored against retained gold labels: on MN-DS's respective $N = 2{,}994$-document sets, and on Newswire's respective $N = 11{,}337$-document sets, reporting the same Micro/Macro/Weighted metrics. Table 5 and Table 6 report the results.

On the single-label dataset (MN-DS, Table 5), our pipeline ($\widetilde{\mathcal{D}}^{\text{SL}}_{\text{train}}$) attains the highest raw Accuracy (0.556), nearly tied with few-shot (0.555). The gap is more pronounced in Macro F1, where our pipeline (0.540) scores 0.17–0.20 points above all random-annotation baselines, driven mainly by higher Macro Recall (0.570 vs. a baseline range of 0.360–0.401) — suggesting the retrieval stage surfaces relevant candidates more evenly across categories rather than favoring majority classes. This mirrors the downstream results of Table 3, where the same pipeline's largest gains are also on Macro F1.

On the multi-label dataset (Newswire, Table 6), $\widetilde{\mathcal{D}}^{\text{ML}}_{\text{train}}$ leads most metrics: Micro Precision (0.710), Micro F1 (0.688), Macro F1 (0.581), and Weighted F1 (0.710). It only lags on Exact Match (0.421 vs. 0.542 for the definition-augmented baseline), likely because LLM-confirmed negatives from the bottom-M pass that disagree with gold labels cause a predicted set to miss an exact match even when only one label is off — a case the all-or-nothing Exact Match penalizes disproportionately relative to Micro or Macro F1. The gap in raw label quality between our method and few-shot (Micro F1 0.688 vs. 0.618) narrows considerably downstream (Table 4), where the trained classifiers reach identical Micro F1 — suggesting the downstream classifier partially compensates for noisier training labels, at least on

Dataset	Accuracy	Macro P	Macro R	Macro F1	Weighted F1
$\widehat{\mathcal{D}}^{\mathrm{SL}}_{\mathrm{ZS}}$	0.525	0.389	0.371	0.347	0.519
$\widehat{\mathcal{D}}^{\mathrm{SL}}_{\mathrm{DEF}}$	0.527	0.410	0.401	0.374	0.521
$\widehat{\mathcal{D}}^{\mathrm{SL}}_{\mathrm{FS}}$	0.555	0.353	0.360	0.337	0.550
$\widetilde{\mathcal{D}}^{\mathrm{SL}}_{\mathrm{train}}$ (Ours)	**0.556**	**0.554**	**0.570**	**0.540**	**0.559**

Table 5
Agreement between the pseudo-labels assigned during dataset construction and the retained gold labels of the same $N{=}2{,}994$ MN-DS documents. Bold values represent the cases with the best agreement between the produced dataset and the ground-truth labels.

	Micro			Macro			Weighted			
Dataset	P	R	F1	P	R	F1	P	R	F1	EM
$\widehat{\mathcal{D}}^{\mathrm{ML}}_{\mathrm{ZS}}$	0.425	**0.710**	0.532	0.396	**0.703**	0.427	0.643	**0.710**	0.579	0.279
$\widehat{\mathcal{D}}^{\mathrm{ML}}_{\mathrm{FS}}$	0.572	0.672	0.618	0.478	0.652	0.492	0.715	0.672	0.622	0.460
$\widehat{\mathcal{D}}^{\mathrm{ML}}_{\mathrm{DEF}}$	0.641	0.663	0.652	0.508	0.619	0.515	0.716	0.663	0.670	**0.542**
$\widetilde{\mathcal{D}}^{\mathrm{ML}}_{\mathrm{train}}$ (Ours)	**0.710**	0.668	**0.688**	**0.613**	0.618	**0.581**	**0.804**	0.668	**0.710**	0.421

Table 6
Agreement between the pseudo-labels assigned during dataset construction and the retained gold labels of the same $N{=}11{,}337$ Newswire documents. Bold values represent the cases with the best agreement between the produced dataset and the ground-truth labels.

aggregate metrics. The advantage of retrieval-guided labels is better preserved on Macro and Weighted F1, the metrics most affected by class imbalance, consistent with the single-label benchmark.

Remark 2. *Notice that the articles in $\widetilde{\mathcal{D}}^{\mathrm{ML}}_{\mathrm{train}}$ are not the same in $\widehat{\mathcal{D}}^{\ell}_{\mathrm{ZS}}$, $\widehat{\mathcal{D}}^{\ell}_{\mathrm{FS}}$, $\widehat{\mathcal{D}}^{\ell}_{\mathrm{DEF}}$, $\widehat{\mathcal{D}}^{\ell}_{\mathrm{GT}}$, for $\ell \in \{\mathrm{SL}, \mathrm{ML}\}$, by construction.*

5. Conclusions and Limitations

Overall, three findings emerge from the performed evaluations. First, training a classifier on pseudo-labeled data consistently beats direct LLM inference on Macro F1, regardless of labeling strategy, with only narrow exceptions on other metrics. Second, our retrieval-and-verification pipeline ranks among the strongest strategies, but its advantage lies less in raw accuracy than in Macro/Weighted F1: on the single-label benchmark it is nearly tied on Accuracy with the best prompting baseline but leads by four points on Macro F1; on the multi-label benchmark it matches the best Micro F1 while leading on Macro and Weighted F1. This suggests the retrieval phase mainly helps by better preserving minority categories rather than by uniformly outperforming simpler strategies. Third, unsupervised methods with carefully engineered category definitions remain surprisingly competitive, especially in single-label settings, offering a low-cost alternative when annotation budget is unavailable.

Finally, three limitations are worth noticing. First, since the unified score $S(x)$ in (3) aggregates relevance into one global ranking, retrieval benefit is not uniform across categories: classes harder to distinguish lexically or semantically can be under-represented in the top-N selection. Future work should explore per-category retrieval and selection thresholds. Second, while our combined sparse/dense/tag-overlap retrieval mechanism proved effective as a proof of concept, alternative combination strategies need deeper study — future work will isolate individual retrieval modules' contributions and explore advanced rank fusion. Lastly, we plan to validate generalizability across new datasets spanning different genres and writing styles, and a wider range of LLMs with more sophisticated prompting strategies, and a cost-accuracy comparison accounting for inference cost rather than LLM call counts alone.

Declaration on Generative AI

During the preparation of this work, the authors used Claude Sonnet 4.6 in order to: grammar and spelling check, and paraphrase and reword. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] A. Bondielli, G. Bonetta, E. Leonardelli, I. Siragusa, Preface to the Ninth Workshop on Natural Language for Artificial Intelligence (NL4AI), in: Proceedings of the Ninth Workshop on Natural Language for Artificial Intelligence (NL4AI 2026) co-located with 24th International Conference of the Italian Association for Artificial Intelligence (AI*IA 2026), 2026.

[2] K. L. Anglin, C. Ventura, Automatic text classification with large language models: A review of openai for zero- and few-shot classification, Journal of Educational and Behavioral Statistics 50 (2025) 1060–1082. URL: https://doi.org/10.3102/10769986241279927. doi:10.3102/10769986241279927.

[3] F. Gilardi, M. Alizadeh, M. Kubli, Chatgpt outperforms crowd workers for text-annotation tasks, Proceedings of the National Academy of Sciences 120 (2023) e2305016120. URL: https://www.pnas.org/doi/abs/10.1073/pnas.2305016120. doi:10.1073/pnas.2305016120. arXiv:https://www.pnas.org/doi/pdf/10.1073/pnas.2305016120.

[4] W. Yin, J. Hay, D. Roth, Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2019, pp. 3914–3923. URL: https://aclanthology.org/D19-1404/. doi:10.18653/v1/D19-1404.

[5] N. Pangakis, S. Wolken, Knowledge distillation in automated annotation: Supervised text classification with LLM-generated training labels, in: D. Card, A. Field, D. Hovy, K. Keith (Eds.), Proceedings of the Sixth Workshop on Natural Language Processing and Computational Social Science (NLP+CSS 2024), Association for Computational Linguistics, Mexico City, Mexico, 2024, pp. 113–131. URL: https://aclanthology.org/2024.nlpcss-1.9/. doi:10.18653/v1/2024.nlpcss-1.9.

[6] M. Mintz, S. Bills, R. Snow, D. Jurafsky, Distant supervision for relation extraction without labeled data, in: Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics (ACL), 2009, pp. 1003–1011. URL: https://aclanthology.org/P09-1113/.

[7] A. J. Ratner, C. M. De Sa, S. Wu, D. Selsam, C. Ré, Data programming: Creating large training sets, quickly, in: NIPS'16: Proceedings of the 30th International Conference on Neural Information Processing Systems, 2016.

[8] A. Ratner, S. H. Bach, H. Ehrenberg, J. Fries, S. Wu, C. Ré, Snorkel: Rapid training data creation with weak supervision, Proceedings of the VLDB Endowment 11 (2017) 269–282. doi:10.1007/s00778-019-00552-1.

[9] D. Chen, A. Fisch, J. Weston, A. Bordes, Reading Wikipedia to answer open-domain questions, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL), 2017, pp. 1870–1879. URL: https://aclanthology.org/P17-1171/. doi:10.18653/v1/P17-1171.

[10] V. Karpukhin, B. Oğuz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, W.-t. Yih, Dense passage retrieval for open-domain question answering, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2020, pp. 6769–6781.

[11] K. Dahiya, N. Gupta, D. Saini, A. Soni, Y. Wang, K. Dave, J. Jiao, K. Gururaj, P. Dey, A. Singh, D. V. Hada, V. Jain, B. Paliwal, A. Mittal, S. Mehta, R. Ramjee, S. Agarwal, P. Kar, M. Varma, Ngame: Negative mining-aware mini-batching for extreme classification, Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining (2022). URL: https://api.semanticscholar.org/CorpusID:250426377.

[12] N. Reimers, I. Gurevych, Sentence-bert: Sentence embeddings using siamese bert-networks, in: Proceedings of EMNLP-IJCNLP 2019, 2019, pp. 3982–3992.

[13] Y. Zhang, M. Li, D. Long, X. Zhang, H. Lin, B. Yang, P. Xie, A. Yang, D. Liu, J. Lin, F. Huang, J. Zhou, Qwen3 embedding: Advancing text embedding and reranking through foundation models, 2025. URL: https://arxiv.org/abs/2506.05176. arXiv:2506.05176.

[14] I. Aarab, Btzsc: A benchmark for zero-shot text classification across cross-encoders, embedding models, and rerankers, in: International Conference on Learning Representations (ICLR) 2026, 2026. URL: https://openreview.net/forum?id=IxMryAz2p3.

[15] P. He, J. Gao, W. Chen, Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing, CoRR abs/2111.09543 (2021). URL: https://arxiv.org/abs/2111.09543. arXiv:2111.09543.

[16] A. Petukhova, N. Fachada, Mn-ds: A multilabeled news dataset for news articles hierarchical classification, 2022. URL: https://doi.org/10.5281/zenodo.7394851. doi:10.5281/zenodo.7394851.

[17] E. Silcock, A. Arora, L. D'Amico-Wong, M. Dell, Newswire: A large-scale structured database of a century of historical news, 2024. URL: https://arxiv.org/abs/2406.09490. arXiv:2406.09490.

[18] A. Singh, A. Fry, A. Perelman, A. Tart, A. Ganesh, A. El-Kishky, A. McLaughlin, A. Low, A. Ostrow, A. Ananthram, A. Nathan, A. L. et al., Openai gpt-5 system card, 2026. URL: https://arxiv.org/abs/2601.03267. arXiv:2601.03267.

[19] S. Bai, Y. Cai, R. Chen, K. Chen, X. Chen, Z. Cheng, L. Deng, W. Ding, C. Gao, C. G. et al., Qwen3-vl technical report, 2025. URL: https://arxiv.org/abs/2511.21631. arXiv:2511.21631.

[20] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized BERT pretraining approach, CoRR abs/1907.11692 (2019). URL: http://arxiv.org/abs/1907.11692. arXiv:1907.11692.