Are We Really Making Much Progress in Text Classification? A Comparative Review

LUKAS GALKE, University of Southern Denmark, Denmark
ANSGAR SCHERP, University of Ulm, Germany
ANDOR DIERA, University of Ulm, Germany
FABIAN KARL, University of Ulm, Germany
BAO XIN LIN, University of Ulm, Germany
BHAKTI KHERA, University of Ulm, Germany
TIM MEUSER, University of Ulm, Germany
TUSHAR SINGHAL, University of Ulm, Germany

We analyze various methods for single-label and multi-label text classification across well-known datasets, categorizing them into bag-of-words, sequence-based, graph-based, and hierarchical approaches. Despite the surge in methods like graph-based models, encoder-only pre-trained language models, notably BERT, remain state-of-the-art. However, recent findings suggest simpler models like logistic regression and trigram-based SVMs outperform newer techniques. While decoder-only generative language models show promise in learning with limited data, they lag behind encoder-only models in performance. We emphasize the superiority of discriminative language models like BERT over generative models for supervised tasks. Additionally, we highlight the literature's lack of robustness in method comparisons, particularly concerning basic hyperparameter optimizations like learning rate in fine-tuning encoder-only language models.

Data availability: The source code is available at https://github.com/drndr/multilabel-text-clf. All datasets used for our experiments are publicly available except the NYT dataset.

CCS Concepts: • **Computing methodologies** → **Supervised learning by classification**; **Neural networks**; • **General and reference** → **Surveys and overviews**; • **Information systems** → *Clustering and classification.*

Authors' addresses: Lukas Galke, University of Southern Denmark, Odense, Denmark, galke@imada.sdu.dk; Ansgar Scherp, University of Ulm, Germany, ansgar.scherp@uni-ulm.de; Andor Diera, University of Ulm, Germany, andor.diera@uni-ulm.de; Fabian Karl, University of Ulm, Germany, fabian.karl@uni-ulm.de; Bao Xin Lin, University of Ulm, Germany, bao.lin@uni-ulm.de; Bhakti Khera, University of Ulm, Germany, bhakti.khera@uni-ulm.de; Tim Meuser, University of Ulm, Germany, tim.meuser@uni-ulm.de; Tushar Singhal, University of Ulm, Germany, tushar.singhal@uni-ulm.de.

CONTENTS

1 INTRODUCTION

1.1 Motivation and Background

Text classification is the task of assigning a categorical label or multiple of such labels to a given text unit [114, 147]. It is a central task in natural language processing, with numerous practical applications, such as classifying scholarly documents, social media posts, news articles, or email spam. Unsurprisingly, until now, text classification has been a very active research field, with new methods appearing every week, as reflected by recent surveys [5, 41, 58, 70, 78, 87, 113, 140, 215, 236].

Besides the rapid pace of research in text classification, conceptually new approaches have also emerged that are not yet sufficiently covered in existing surveys. Those include the use of graph neural networks (GNNs) to process text, which attracts increasing attention from researchers, and the rise of large language models (LLMs), which are often (naively) assumed to be the state-of-the-art in all-natural language processing tasks. Although there are surveys with good coverage of new methods, e. g., focusing on LLMs [58] or GNNs [95], those then lack a quantitative comparison. Other surveys perform a quantitative comparison but focus on single-label classification and use their own splits instead of the established splits per dataset [142]. This survey covers both of these new families alongside classical approaches and provides a quantitative comparison across multi-class (or single-label), multi-label, and hierarchical text classification.

Many new methods for text classification are based on graph neural networks (GNNs) [52]. Common to these GNN-based approaches is that they first generate a synthetic graph from the corpus that contains text-augmented vertices and edges with information on word and document co-occurrences, e. g., [200, 229]. Second, the GNN is trained on this graph to carry out the text classification task. In hierarchical text classification, where classes are organized along a thematic hierarchical thesaurus [147], GNNs are often used to encode the label hierarchy, e. g., [66, 183].

Another group of text classification methods is based on language models, which can be organized in encoder-only, encoder-decoder, and decoder-only models [197].[1] Encoder-only pre-trained language models such as BERT [31] took big strides in many natural language processing tasks including categorical text classification [43, 45]. The encoder-only transformer language models were followed by encoder-decoder variants T5 [138] and decoder-only generative large language models (LLMs) such as the GPT models- [13, 121]. Decoder-only transformer language models focus on text generation with remarkable in-context learning abilities. This makes them strong zero-shot and few-shot models [13, 127, 158, 185], yet whether in-context learning LLMs are superior to fine-tuned small language models is highly questionable [14, 38, 83, 92, 96, 136].

Decoder-only language models are usually much larger than masked language models [13], which is because the causal language model objective architecture (left-to-right prediction) allows for easier scaling through diagonal masking of the attention matrices – enabling the model to get an error signal for each token in a batch. Thus, we increasingly find the distinction between SLMs (small language models) and LLMs, e. g., [8, 56, 155], which is based on a (kind of arbitrarily) chosen parameter count. We argue that a key distinction between SLMs and LLMs, particularly for classification tasks, lies in the pre-training objective. The encoder-only and encoder-decoder SLMs (e. g., BERT and T5) are based on masked language modeling (with subsequent fine-tuning of a discriminative classifier in BERT and its variants) while the decoder-only LLMs (e. g., GPTs) are

---

[1]Disclaimer: We are well aware of the vivid discussions and evolving organization of language models, e. g., on social media like LeCun's post https://twitter.com/ylecun/status/1651762787373428736 that encoder-only models of course also have a decoder, but "just not an auto-regressive decoder", etc. We follow the key distinction between language models as discussed by LeCun and surveys such as Yang et al. [197]. We focus on the central aspect of distinguishing language models for classification tasks, i. e., between fine-tuned discriminative models versus generative large language models as this distinction, central to our observations in the survey, as it also corresponds to whether task-specific fine-tuning is employed.

pre-trained using the causal language modeling objective. This side-effect of pre-training objectives is not by coincidence but rather an effect of the causal language model objective providing a training signal for each token, whereas the masked language model objective only provides a training signal for each *masked* token.

Another side effect of scale is that SLMs are usually fine-tuned for a specific downstream task. Although task-specific fine-tuning is also possible with LLMs through techniques like Low-Rank Adaptation [57], it is more prohibitive due to the higher number of parameters with common model, such that most approaches leveraging LLMs rely on more general instruction fine-tuning [122] followed by in-context learning with few examples and dedicated prompting schemes [158].

1.2 What is the Problem? Why is our Study Needed?

Encoder-only language models like BERT have led to major improvements to the state of the art on many NLP tasks, including categorical text classification [43, 45]. Our question here is whether the numerous newly proposed methods provide substantial improvements over encoder-only language models. We pinpoint this question to three aspects analyzed in this paper, namely the apparent ineffectiveness of using synthetic graphs, the importance of task-specific fine-tuning, even in the era of LLMs, and the general challenges of fairly assessing new methods and baselines.

- *Synthetic Graphs* The use of GNNs has been specialized to the task of text classification by synthetic graphs induced from the text corpus, e. g., TextGCN [200]. However, those synthetic graphs may not provide information beyond what a neural network can already directly extract from the text [45].
- *Large Language Models* The literature suggests that generative LLMs do not generally improve over encoder-only SLM for text classification tasks [90, 158, 209, 213, 228]. This comes despite the strong advantages of generative LLMs in their sheer size of parameters as well as improvements based on techniques like prompt engineering [158, 188], instruction fine-tuning [185], and prompt-tuning [84, 102].
- *Comparability of Methods* A general challenge when assessing the performance of methods is the comparability of the results. Besides properly reporting the used datasets, splits, preprocessing, etc. an important question is if baselines are properly optimized [26, 81]. It is especially challenging to properly compare methods for text classification when so many new papers appear using GNNs and language models.

1.3 Methodology

We extensively review the literature in the field of modern and classical machine learning methods for single-label and multi-label text classification. Based on the literature search, we derive the families of methods for text classification.

- Methods based on Bag of Words (BoW) using, e. g., a support vector machine or a multi-layer perceptron [45].
- Methods that consider text as a sequence of tokens such as the encoder-only BERT [31] or the decoder-only GPT [13],
- Graph-based methods that employ graph neural networks on synthetic graphs like TextGCN [200] and hierarchy-based text classification methods like HGCLR [183].

We determine established single-label and multi-label benchmark datasets, which we consider in our comparison, and identify the top-performing methods. We carefully probe the validity of the results for each method and paper found in the literature. We check, among others, the train-test split used (whether they deviate from established benchmark splits), the number of classes considered, the hyperparameter values (are they provided and comparable, whether the baselines

are optimized, the metrics applied, and if there is any unusual preprocessing of the datasets that may have influenced the results.

We aggregate all results found in the literature. We identified gaps in the use of methods and datasets, i.e., when certain combinations of models and datasets could not be found. Where needed, we run own experiments to fill gaps. Overall, we achieve a systematic comparison of the different text classification methods among the different families of methods.

1.4 Key Results

The family of fine-tuned transformer language models defines the state of the art for single-label and multi-label text classification tasks. Despite recent advances in LLMs, the best-performing models for text classification are still SLMs BERT and its variants RoBERTa [105] and DeBERTa [54]. Despite their sheer amount of parameters and larger pretraining, methods based on in-context learning with LLMs do not generally outperform fine-tuned SLMs in text classification. Even when pushing the limit of using such generative language models by applying tricks such as advanced prompting techniques [158] or using ensembles [228], the performance does not, or only marginally on individual datasets, outperform those of encoder-only SLMs. For example, it requires an ensemble of Llama LLMs to reach or slightly outperform an encoder-only SLM on two benchmark datasets [228]. We attribute these observations to two main factors. First, task-specific fine-tuning is still an important distinctive criterion. The encoder-only models like BERT are pre-trained with the masked language modeling objective and fine-tuned with a classifier head for specific datasets on the text classification task. Second, the left-to-right attention mask is suboptimal for text classification tasks. Therefore, fine-tuned models with unrestricted attention are effective text classifiers and should be preferred over purely generative models for text classification tasks. This again relates well to the distinction between generative and discriminative approaches [116] – the former aiming to learn the full joint distribution $p(x, y)$, which we interpret here as in-context learning, and the latter focusing on the decision boundary directly, i.e., $p(y|x)$, which we interpret here as task-specific fine-tuning.

For the GNN-based methods, despite the huge number of methods developed in recent years, the idea of exploiting a synthetically induced graph to improve text classification has not lived up to its promise. Many graph-based methods such as [103, 139, 200] are not only outperformed by simply applying BERT but already fall below simple baselines such as a logistic regression or a simple classifier such as a multi-layer perception on a bag-of-words representation [45, 139]. Such baseline models already extract relevant information for text classification from the raw text [147]. Adding a synthetically generated graph from the corpus does not provide additional information that a neural network can exploit [45].

A worrying observation is the lack of strong baselines and/or not properly tuning them appropriately. Like Dacrema, Cremonesi, and Jannach [26] observed for neural recommender systems that "works can be outperformed at least on some datasets by conceptually and computationally simpler algorithms", the reason is also that baselines are not properly optimized [149] ("everyone's a winner"). Classical machine learning methods such as support vector machines, logistic regression, and multi-layer perceptrons are largely ignored in papers published in the last years, despite showing consistently strong results across various datasets [43, 45, 139]. There are cases where even the baseline's most basic hyperparameter, the learning rate (or fine-tuning learning rate, in the case of BERT models) is not properly considered. We show this in the example of BERT and its variants, which makes these models perform considerably worse compared to what they can achieve. The literature shows a quite high discrepancy in BERT's performance on benchmark datasets, up to a 13 points difference in accuracy, which can be attributed to the choice of the fine-tuning learning rate.

Following studies in other areas of machine learning, we conclude that strong baselines must be used and their hyperparameters properly optimized as it is an important means to argue about *true* scientific advancement [26, 150], and also their usefulness in practical settings.

1.5 Remainder

The remainder of this article is organized as follows: Below, we introduce our survey methodology. We survey the literature along the families of methods for text classification, beginning with BoW-based models in Section 3, sequence-based methods in Section 4, and concluding with graph-based methods in Section 5. The experimental apparatus, including the considered datasets, and the procedure for running our own experiments to fill gaps in the literature is described in Section 6. The results of our quantitative comparison are presented in Section 7. Subsequently, we discuss limitations in Section 9, findings in Section 8, and promising future directions in Section 10, before concluding.

2 METHODOLOGY

This survey covers single-label text classification, multi-label text classification, and hierarchical text classification – covering published methods up to January 2025. We do not consider *extreme* multi-label classification, where the focus is dealing with very large label space, which requires a different set of specialized techniques [73, 74, 205, 219] and different metrics (ranking metrics rather than classification metrics), whose comparison we leave for future work.

In a first step, we have collected and analyzed recent surveys on single-label and multi-label text classification and searched for research papers that include comparison studies [5, 15, 37, 44, 45, 70, 78, 87, 101, 109, 113, 135, 140, 165, 177, 218, 236]. These cover the range from classical machine learning models to deep classification models. Second, we have screened for literature in key NLP and artificial intelligence venues. Finally, we have complemented our search by checking results and papers on https://paperswithcode.com/task/text-classification (for single-label) and https://paperswithcode.com/task/multi-label-text-classification (for multi-label).

Based on this input, we have determined the families of methods and benchmark datasets used (see Table 1). This categorization is mainly based on the method signature, distinguishing whether methods operate on BoW, sequence, or graph-structured input [45]. We extend the categorization by further distinguishing between two types of graph-based methods: First, approaches that employ a graph structure derived from a corpus of text documents, which we call synthetic text-graph approaches. Second, hierarchy-based methods that are using a graph model to encode the hierarchical structure among classes. We further add the subcategories of large language models and attention-free language models to the family of sequence-based models. Figure 1 shows an overview of the categorization.

We focus our analysis on methods that show strong performance and include them in our study. We have verified that the same train-test split is used for all methods. We check whether modified versions of the datasets have been used (e. g., fewer classes) to avoid bias and mistakenly give advantages.

We do not consider papers that do not allow for a fair comparison with the state of the art. Reasons include that they

- used different or non-standard benchmark datasets only like [1, 12, 17, 19, 24, 27, 46, 47, 51, 58, 63–65, 90, 100, 107, 118, 131, 133, 133, 153, 157, 162, 166, 167, 170, 186, 193, 196, 209–211, 213, 226, 233, 238, 239],
- modified the datasets to use a different number of classes as done in [79, 126, 235],
- employed different train-test splits like [16, 34, 38, 115, 123, 140, 142, 156, 160, 175],
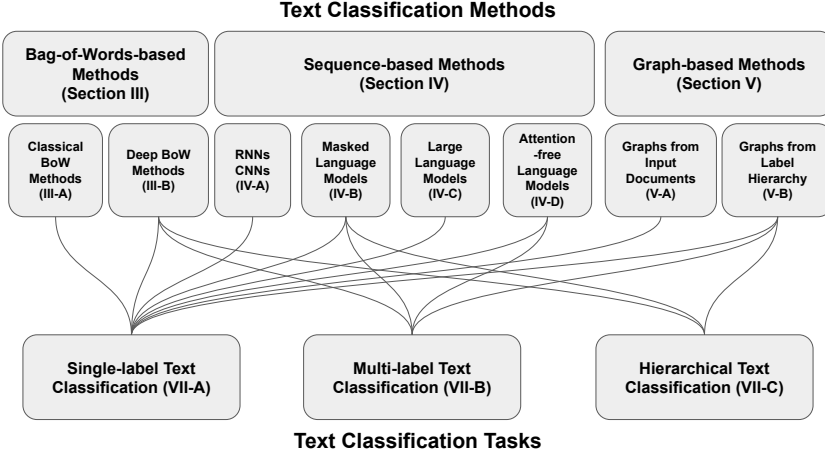
Text Classification Methods

Fig. 1. Categorization of text classification methods in families (top) and tasks (bottom). We draw a line if at least one of the combinations of method and task is covered in our quantitative comparison.

- train-test splits are not reported [115, 146, 214],
- used fewer training examples [37, 93, 99, 160, 172, 182, 192, 231, 238], or
- used different evaluation measures [61, 163].

The rationales for why certain changes are made were not always clear in the literature. However, it reflects a general problem of comparability in machine learning research [81].

Below, we describe the family of BoW methods in detail in Section 3, sequence-based methods in Section 4, and graph-based methods in Section 5.

3 BOW-BASED METHODS

Under pure BoW-based (Bag of Words) text classification, we denote approaches that operate only on a multiset of words (or tokens) from the input document. Given paired training examples $(x, y) \in \mathcal{D}$, each consisting of a vector that holds the frequency of words in its components $x \in \mathbb{R}^{n_{\text{vocab}}}$, which is commonly referred to as a bag (multiset) of words, and a class label $y \in \mathbb{Y}$. The goal is to learn a generalizable function $\hat{y} = f_\theta^{(\text{BoW})}(x)$ with parameters θ such that $\arg\max(\hat{y})$ is the true label y for input x. For multi-label classification, the BoW-based model considers multiple class labels. Instead of using $\arg\max(\hat{y})$ to decide on a label, a binary sigmoid output is commonly used per label and, along with a threshold λ that determines whether the corresponding class will be assigned. The multi-label model is trained with a binary cross-entropy loss instead of a categorical cross-entropy. As output, the multi-label classifier can produce between 0 to $|\mathbb{Y}|$ many labels [147], i. e., it is possible that no label is predicted.

3.1 Classical BoW Methods

Classical machine learning methods that operate on a BoW-based input are extensively discussed in surveys and comparison studies [44, 70, 78, 139]. These studies show that the best-performing classical models are Support Vector Machines (SVM) and logistic regression (LR). Especially, the strong performance of logistic regression is astonishing. For instance, Ragesh et al. [139] have shown that logistic regression outperforms the advanced graph-based TextGCN [200] method.

3.2 Deep BoW Methods

With more advanced BoW methods, Galke et al. [44] have found that an MLP on a bag-of-words representation of the text outperforms many graph-based approaches. Earlier approaches are mainly based on pre-trained word embeddings [112, 132]. For instance, Iyyer et al. [62] proposed Deep Averaging Networks (DAN), a combination of word embeddings and deep feedforward networks. DAN is an MLP with one to six hidden layers, non-linear activation, dropout, and AdaGrad as an optimization method. The results suggest that pre-trained embeddings such as GloVe [132] would be preferable over randomly initialized neural bag-of-words [71]. In fastText [11, 69], a linear layer is used on top of pre-trained embeddings for classification. Furthermore, Henao et al. [150] explore different pooling variants for the input word embeddings and find that their Simple Word Embedding Models (SWEM) can rival approaches based on recurrent (RNN) and convolutional neural networks (CNN). Note that approaches such as fastText and SWEM that apply a logistic regression on top of pre-trained word embeddings share a similar architecture as an MLP with one hidden layer. However, the standard training protocol involves pre-training the word embedding on large amounts of unlabeled text and then freezing the word embeddings while training the logistic regression [112].

4 SEQUENCE-BASED METHODS

As sequence-based methods, we consider recurrent and convolutional neural networks, SLMs (e. g., BERT), LLMs (e. g., GPT-3.5), and attention-free language models (e.g., gMLP). These approaches aim for *contextualized* word representations, for which nearby words and word order are taken into account. The model signature for sequence-based methods is $\hat{y} = f_\theta^{(\text{sequence})}(\langle x_1, x_2, \ldots, x_k \rangle)$, where k is the (maximum) sequence length.

4.1 Recurrent and Convolutional Neural Networks

Before the era of pre-trained language models, recurrent neural networks (RNN) were a natural choice for any NLP task. SGM is a model for multi-label classification based on a bidirectional LSTM with an attention mechanism [198], which later has been extended to use T5 as a backbone language model [207]. BLSTM-2DCNN [235] is a bidirectional LSTM with two-dimensional max pooling. It has been applied to a subset of the 20ng dataset with four classes. Thus, the high score of 96.5 reported for 4ng cannot be compared with papers applied to the full 20ng dataset, such as [45]. Also TextRCNN [79], a model combining recurrence and convolution uses only the four major categories in the 20ng dataset. The results of TextRCNN are identical to BLSTM-2DCNN. For the MR dataset, BLSTM-2DCNN provides no information on the specific split of the dataset. RNN-Capsule [181] is a sentiment analysis method reaching an accuracy of 83.80 on the MR dataset but with a different train-test split. Lyu and Liu [108] combine a 2D-CNN with bidirectional RNN. Another work applying a combination of a convolutional layer and an LSTM layer is by Wang et al. [179]. The authors experiment with five English and two Chinese datasets, which are not in the set of representative datasets we identified. The authors report that their approach outperforms existing models like fastText on two of the five English datasets and both Chinese datasets.

4.2 Small Language Models

The transformer [173], originally developed for machine translation, introduced a key-value self-attention mechanism, and had an immense impact on language modeling. transformer-based language models are pre-trained with a self-supervised training objective on large text corpora and can be subsequently fine-tuned with a supervised objective on a specific task. Generally, transformer-based language models can be categorized into encoder-only models, encoder-decoder

models, and decoder-only models [197]. BERT and its follow-up approaches are encoder-only language models and are trained with the masked-language modeling objective [31]. Their main purpose is encoding the text and performing a classification or regression task with a dedicated module (e. g., a linear output layer) on top of the encoder. The output layer is trained from scratch (i. e., from random initialization) during fine-tuning.

Encoder-decoder language models such as T5 [138] cast all downstream tasks, including classification, into a text-to-text framework with task-specific prompt prefixes. The rationale is that this practice enables multi-task fine-tuning with the aim of similar tasks improving each other. Although these models generate text, they can also be used for classification tasks by interpreting generated tokens as class labels. In multi-label classification, text-to-text (= sequence-to-sequence) models have been used even before the era of large language models [117] to facilitate the prediction of multiple labels.

Popular follow-up works of BERT are RoBERTa [105], DistilBERT [145], ALBERT [80], De-BERTa [54], and ERNIE 2.0 [159]. RoBERTa improves the pre-training procedure of BERT and removes the next sentence prediction objective. DeBERTaV3 [53] is a modified DeBERTa with an ELECTRA-style pre-training [22]. DistilBERT [145] is a distilled version of BERT with 40% reduced parameters and 60% faster inference time that retains 97% of BERT's performance on the GLUE benchmark. ALBERT reduces the memory footprint and increases the training speed of BERT for improved scalability. Like DistilBERT and ALBERT, TinyBERT [67] and MobileBERT [161] are also size-reduced variants of BERT, but these two need the original BERT model for fine-tuning. DeBERTa introduces a disentangled attention mechanism, i. e., keeping word position and content embeddings separate, and an enhanced mask decoder, which introduces absolute position encodings to the final softmax. ERNIE 2.0 employs a continual multi-task learning strategy. Whenever a new task is introduced, the previous model parameters are used for initialization, and the new task is added to the multi-task learning objective. Finally, ModernBERT is a refurbished variant of the BERT model integrating various optimizations developed over the years and providing an input length of 8, 192 tokens [184].

Pre-trained language models have also found their way into hierarchical text classification. For instance, HBGL [66] uses BERT to represent the text and represent the hierarchically organized classes. For the classes, HBGL first learns the label embeddings from the global hierarchy, i. e., the taxonomy. Specifically, the label embeddings are learned by employing a masked language modeling objective on sequences that correspond to paths in the label hierarchy. Subsequently, it learns to predict the document labels one by one in the order of the local hierarchy, i. e., level-wise from the root to the most specific label in the taxonomy. By this, HBGL exploits the hierarchy of the labels as defined in the taxonomy and treats the label generation as a multi-label text classification task in a similar way as Nam et al. [117] did with sequence-to-sequence models trained from scratch. Thus, HBGL is essentially a *hierarchy-aware* sequence-based transformer model. However, as described above, the key ingredient of HBGL is making use of the sequence-based model BERT. HBGL *does not use an external graph encoder* for representing the taxonomy. In contrast, the graph-based methods (described in Section 5) always use an *explicit* graph encoder for representing the taxonomy, most commonly a graph neural network, independent of what other model is being used (e. g., a CNN, BERT, or other). Since HBGL's core is heavily based on BERT and does not have an explicit graph encoder, we consider it a sequence-based transformer method.

The encoder-decoder model Seq2Tree adopts T5 to perform hierarchical text classification [207]. It uses a depth-first search approach to linearize the label hierarchy in order to encode it as a sequence in the model. Finally, RADAr is a sequence-to-sequence model that uses RoBERTa as a text encoder and a custom two-layer Transformer-based decoder for hierarchical text classification [206]. In contrast to hierarchical text classification methods, see Section 5.2, RADAr does not operate on

a given dataset taxonomy. Beyond using ROBERTa as an encoder in RADAr, we also experimented with order transformer models, including BERT, XLNet, and DeBERTa. The results were generally very similar, with RoBERTa achieving the slightly highest scores overall.

Other approaches that use small language models for text classification include retrieval and data augmentation from an external knowledge base, such as Zhu et al. [238], who query ProBase to improve short text classification with BERT. For reasons of a fair comparison, we do not consider approaches that make use of an external knowledge base in our quantitative comparison.

4.3 Large Language Models

Decoder-only language models such as GPT-3 [13] are trained with a left-to-right, i. e., causal language modeling, objective. This makes decoder-only models most suitable for text generation. However, decoder-only models are very flexible and can also be used to carry out other downstream tasks, including text classification [13?]. This is done by specifying the task in natural language and providing a few examples in the prompt [13]. This practice, known as in-context learning, does not require updating the model. In-context learning has led to increased interest in the design of and working with prompts, such as Chain-of-Thought (CoT) prompting [187] *inter alia.*

The state-of-the-art prompting technique for text classification is Clue And Reasoning Prompting (CARP) [158], where the instructions consist of first finding relevant clues in the text input and then providing a classification result based on the clues along with an explanation.

The recent work by Sun et al. [158] evaluates GPT-3.5+CoT on text classification and introduces a prompting strategy called Clue and Reasoning Prompting (CARP). The authors evaluate GPT-3.5, GPT-3.5+CoT, and GPT-3.5+CARP with two different samplers for selecting the in-context examples. The samplers are a uniform sampler and a RoBERTa model fine-tuned for the current downstream task. Using RoBERTa representations of the training documents, the sampler employs a kNN search on the examples to sample more representative documents per class to be included in the prompt. The best-performing CARP variant employs a 16-shot RoBERTa sampler with a majority vote over multiple runs of prompting GPT-3.5, which we denote as GPT-3.5+CARP+vote.

Specifically for text classification, prompt boosting has shown promising results [55], where differently prompted LLMs were ensembled with an adaptive boosting algorithm. However, the few-shot performance of large language models without any fine-tuning is still lower than fine-tuned small language models [38]. Beyond prompting techniques, Zhang et al. [228] have experimented with fine-tuning an ensemble of Llama-2 [169] models for text classification leading to competitive scores. Li et al. [92] do report promising results when fine-tuning a Llama-2 model [169] without causal masking such that both left and right context can be considered during self-attention.

In the end, it is currently unknown whether in-context learning with a large language model is sufficient for a given task or whether fine-tuning is needed.

4.4 Attention-free Language Models

The self-attention mechanism has been very successful, but it has quadratic complexity in sequence length. After transformer-based models have also entered the vision domain [36], Google researchers introduced methods that eliminate the costly self-attention mechanism in transformers and are purely based on MLP layers. The first of these attention-free models is MLP-Mixer [168] developed for vision tasks. It divides the input image into a sequence of non-overlapping patches, then fed through blocks of MLPs consisting of channel-mixing and token-mixing layers.

Shortly after releasing the MLP-Mixer architecture, an MLP-based natural language processing model called gMLP [97] was released. The gMLP model replaces the attention layer in the basic blocks of a transformer with a spatial gating unit. Inside this layer, cross-token interactions are achieved by multiplying the hidden representation element-wise and projecting it linearly.

While Liu et al. [97] found that it is possible to achieve similar performance as BERT by replacing self-attention with these gating units, gMLP was still outperformed by BERT on some tasks. The authors hypothesized that self-attention could be advantageous depending on the tasks (i. e., cross-sentence alignment). Therefore, they attached a tiny attention unit (single-head with size 64) to the gating units. This extension is called aMLP and substantially increases the model's performance. While other attention-free language models have been proposed [49, 129], none of them has been systematically evaluated for topical text classification.

5 GRAPH-BASED METHODS

Graphs can serve several purposes in text classification: One is to consider the input data as a graph (i. e., the documents, their words), which we call synthetic text-graph approaches to distinguish them from graphs in which the structure has a natural interpretation (e. g., citations graphs). Another purpose is considering an additional label hierarchy as input to the model [147] (i. e., classes organized in a hierarchy), which we call hierarchy-based methods.

5.1 Synthetic Text-Graph Methods

Using graphs induced from text has a long history in text classification. An early work is the term co-occurrence graph of the KeyGraph algorithm [120]. The graph is split into segments, representing the key concepts in the document. Co-occurrence graphs have also been used for automatic keyword extraction [143] and classification [221]. Modern methods exploit this idea of a graph induced from the text. The text corpus is first transformed into a graph, which is then fed as input into a graph neural network (GNN) [52].

The synthetic text-graph approaches to text classification first set up a *synthetic* graph based on the text corpus $\mathcal{D}$ such that an adjacency matrix is created from a document corpus $\hat{A} :=$ make-graph($\mathcal{D}$). The graph is composed of word nodes and document nodes, each receiving its own embedding (by setting $X = I$). For example, in TextGCN, the graph is created from word-word edges (modeled by pointwise mutual information) and word-document edges (resembling word occurrence in the document). Then, a parameterized function $f_\theta^{(\text{graph})}(X, \hat{A})$ is learned that uses the graph as input, where X are the node features. Note that graph-based approaches such as TextGCN disregard word order, similar to the BoW-based models described above.

Among others, methods that follow this synthetic text-graph approach include TextGCN [200], TensorGCN [103], HeteGCN [139], HyperGAT [32], HGAT [199], DADGNN [103], STGCN [201], SHINE [182], AGGNN [29]. While many of these methods are strictly transductive such as HeteGCN and TensorGCN, other methods are inductive [59, 60, 178, 229] or can be adapted to become inductive [139]. Transductive models need access to the unlabeled test documents at training time. This requires computing the graph also on the documents of the test set and making this information available during training (but without the labels from the test set). In contrast, inductive models can be applied to new data. Here, the graph induced from the text is computed only on the training set.

Transductive training has inherent drawbacks as the models cannot be applied to new documents. For example, in TextGCN's original transductive formulation, the entire graph, including the unlabeled test set, must be available for training. This may be prohibitive in practical applications as each batch of new documents would require retraining the model. When TextGCN and other graph-based methods are adapted for inductive learning, where the test set is unseen, they achieve notably lower scores [139]. Note that all previously described bag-of-words and sequence-based models fall in the inductive category and can be applied to new documents.

We briefly discuss selected graph-based methods. In TextGCN, the authors set up a graph with word and document nodes. Word–word edges are derived from pointwise mutual information (PMI) and word–document edges are derived from TF-IDF scores. This synthetic graph is then fed into a graph convolutional network (e. g., a GCN [76]) with the goal of classifying the document nodes. HeteGCN combined ideas from Predictive Text Embedding [164] and TextGCN and splits the adjacency matrix into its word-document and word-word sub-matrices and fuse the different layers' representations when needed. TensorGCN explores and combines multiple different ways of converting the text into a graph, such as a semantic graph created with an LSTM, a syntactic graph created by dependency parsing, and a sequential graph based on word co-occurrence. HyperGAT combines graph attention [174] with the concept of hyperedges based on sequential structure and topic models [10]. AGGNN [29] focuses on text pooling mechanism along with gated graph sequence neural networks [91]. DADGNN is a graph-based approach that uses attention diffusion and decoupling techniques for tackling the over-smoothing problem of the GNN and building deeper models. Lastly, STGCN tackles short text classification by building upon ideas from TextGCN and adding word-topic and document-topic edges from a topic model, similar to HyperGAT. The authors also experimented with combining STGCN with a BiLSTM and a BERT model. In their experiments, the combination STGCN+BERT+BiLSTM gave the best results, while pure STGCN fell behind pure BERT. MHGAT [68] follows a different approach and captures word order by adding position-specific hyperedges to the graph to be processed by a graph attention network. These position edges are obtained through a sine/cosine transformation, following a similar strategy as in the Vaswani transformer's positional encoding [173].

Particularly between 2022 and 2024, numerous new graph-based models have been published. Many of them use BERT in conjunction *with an explicit graph encoder*, usually a graph neural network, such as in BertGCN [94], CTGCN [194], ILGCN [152], TSW-GNN [89], and ConTex-tING [60]. They differ from the other GNN-based methods as the graph is not computed based on word co-occurrences but BERT's subword tokens. Further graph-based methods are KGAT [180], InducT-GCN [178], TextSSL [134], GLTC [203], and others. A recent survey on GNNs for text classification was performed by Wang, Ding, and Han [177]. Moreover, Bugueno and de Melo [15] compare different initial document representations including Word2vec [112], GloVe [132], and frozen BERT embeddings and use them in conjunction with graph neural networks. They employ frozen and fine-tuned BERT models as baselines for the categorical text classification task. Their results show that—on most datasets—graph neural networks hardly compete with a fine-tuned BERT.

5.2 Hierarchy-based Methods

Apart from the text-induced graphs used by the methods described above, also the classes of the dataset may be organized in a graph structure. This is typically the case in hierarchical text classification, where each document should be annotated with a set of labels rather than a single class label [147]. and the classes are organized along a taxonomy. The taxonomic hierarchy of labels is typically modeled as a tree or a directed acyclic graph [130, 151, 232]. The goal is then to predict multiple class labels which correspond to one or more nodes in the hierarchy.

Following the taxonomic hierarchy to the root, the classes become more general (broader), while going towards the leaves, the classes become more specific (narrower). The documents are typically annotated with some specific classes in the taxonomy. However, in hierarchical text classification, this taxonomy is often used to *enrich* the gold standard [232]. This means that all vertices along the entire path from the root to the assigned classes are added as ground truth. The hierarchy-based text classifier HiAGM [232] and its follow-up works, such as HGCLR [183] and HBGL [66], rely on this enrichment.

The enrichment affects both the training and evaluation of hierarchy-aware methods. After this enrichment, the dataset consists of a set of documents X. Each document is annotated with multiple labels, typically modeled as a label indicator matrix $\hat{Y}$, and a hierarchy of classes H. Then, the goal is to learn a function $x, H \mapsto \hat{y}$ that maps the current document x to a set of enriched labels $\hat{y}$, while also taking into account the label hierarchy H. The hierarchy-based methods make use of an explicit graph encoder to represent the taxonomy and to exploit it in the model architecture to classify the text.

We briefly discuss the selected methods. HiAGM is a hierarchical text classifier that models the hierarchy as a directed graph along with hierarchy-aware structure encoders [232]. It comes in two variants: HiAGM-LA, which is a multi-label attention model that uses an inductive approach. HiAGM-TP is a text feature propagation model that uses a deductive approach to extract hierarchy-aware text features. It uses a GNN-based encoder to obtain a representation for each class and compare it with a Tree-LSTM representation of the text. In early 2021, several other hierarchical label-based attention models were published. For example, HLAN [35], LA-HCN [225], RLHR [98], and the weakly-supervised TaxoClass [151]. Further, hierarchy-based methods are [191, 204, 222]. We consider HiAGM in our comparison as well as methods that use BERT for text and a graph neural network for the hierarchy, such as BERT+HiMATCH [20], as well as HGCLR, which uses contrastive learning to align text and graph representations. K-HTC combines BERT as a text encoder with a knowledge graph based on entities relevant to the classification task [106].

The hierarchy-aware and label-balanced (HALB) model extends HGCLR by replacing the classification with asymmetric loss and adds another loss for separating samples with similar representation but different labels [220]. The Hierarchy-aware Information Lossless contrastive Learning (HILL) model uses BERT as a text encoder and a graph encoder together with a hierarchy-aware contrastive loss [237]. HGBL is another model based on contrastive learning on the label and text features using a graph-encoder and BERT as text encoder [217].

The methods discussed so far rely on a small encoder-only language model. Retrieval-style ICL is an approach for hierarchical text classification using a large-language model and few-shot in-context learning [21].

6 EXPERIMENTAL APPARATUS

Here, we introduce the benchmark datasets we identified for the single-label and multi-label text classification tasks. We provide an overview of the models considered from the different families of text classification approaches and indicate where we add own experiments to fill gaps. We describe our procedure, choice of hyperparameters and their optimization, and evaluation measures.

6.1 Datasets

Our quantitative comparison focuses on topic classification, while including popular sentiment analysis datasets as control: MovieReviews for single-label classification, and GoEmotions for multi-label classification. We include five single-label and seven multi-label datasets described in the following.

Single-label Datasets. We use the benchmark datasets 20ng, R8, R52, ohsumed, and MR with their standard train-test splits. Twenty Newsgroups (20ng)[2] (bydate version) contains long posts categorized into 20 newsgroups. R8 and R52 are subsets of the R21578 news dataset with 8 and 52 classes, respectively. Ohsumed[3] is a corpus of medical abstracts from the MEDLINE database that

---

[2]http://qwone.com/~jason/20Newsgroups/

[3]http://disi.unitn.it/moschitti/corpora.htm

are categorized into diseases (one per abstract). Movie Reviews (MR)[4] [125], split by Tang et al. [164], is a binary sentiment analysis dataset on sentence level. Table 1 shows the dataset characteristics.

Table 1. Characteristics of the single-label text classification datasets. We show the number of documents N and the standard train-test split. #C is the number of classes. Finally, we report the documents' average length and standard deviation.

Dataset	N	#Train	#Test	#C	Avg. length
20ng	18,846	11,314	7,532	20	$551 \pm 2{,}047$
R8	7,674	5,485	2,189	8	119 ± 128
R52	9,100	6,532	2,568	52	126 ± 133
ohsumed	7,400	3,357	4,043	23	285 ± 123
MR	10,662	7,108	3,554	2	25 ± 11

Multi-label Datasets. Table 2 shows the characteristics of the multi-label datasets. Reuters-21578 (R21578) [3] is a popular dataset for multi-label classification. It is a collection of documents that appeared on Reuters newswire in 1987. We use the train-test split from NLTK.[5] The labels in R21578 are not hierarchically organized. RCV1-V2 is a newer version of the R21578 dataset containing a much larger amount of hierarchically categorized newswire stories. For RCV1-V2, we use the train-test split proposed by Lewis et al. [85]. EconBiz [109] is a dataset containing scientific papers in economics. It provides the titles of a meta-data export as well as the full text of papers up to 2017. EconBiz does not provide a specific train/test-split, but the samples are split into eleven parts. Parts 0 to 9 correspond to the documents with titles and full text, while part 10 contains papers where only the titles are available. This organization of the dataset is due to the research question addressed by Mai et al. [109] comparing text classification using full-text versus only employing the titles. In order to accommodate this dataset in our experiments, we use the titles from part 10 for training and the titles from parts 0–9 documents for testing.

GoEmotions is a corpus of comments extracted from Reddit, with human annotations to 27 emotion categories [28]. We use the same train-test split as in the original paper. GoEmotions does not have a hierarchical label structure. Amazon-531 [111] contains 49,145 product reviews and a three-level class taxonomy consisting of 531 classes. DBPedia-298 [82] includes 245,832 Wikipedia articles and a three-level class taxonomy with 298 classes. For Amazon-531 and DBPedia-298, we use the same train-test split as in TaxoClass [151]. NYT AC [144] contains New York Times articles written between 1987 and 2007. We use the train-validation-test split from HiAGM [232]. In the two datasets NYT and RCV1-V2, each label set includes the more general labels along the path up to the root of the hierarchy, i. e., their label sets are enriched as it is commonly done in the literature on hierarchical text classification.

6.2 Methods and Complementing Experiments

We build our quantitative comparison on existing studies such as Ding et al. [32], Ragesh et al. [139], Li et al. [86] and Galke et al. [45]. Where needed, we fill gaps in the literature by running own experiments.

Below, we describe the considered models along the families introduced in Sections 3, 4 and 5.

---

[4]https://www.cs.cornell.edu/people/pabo/movie-review-data/
[5]https://www.nltk.org/book/ch02.html

Table 2. Characteristics of the multi-label classification datasets. We show the same statistics for the single-label datasets. In addition, we report the average number of class labels per document and whether the dataset comes with a hierarchy (Hier.).

Dataset	Hier.	N	#Train	#Test	#C	Labels per doc.
R21578	N	10,788	7,769	3,019	90	1.24 ± 0.75
RCV1-V2	Y	804,414	23,149	781,265	103	3.24 ± 1.40
EconBiz	Y	1,064,634	994,015	70,619	5,661	4.36 ± 1.90
GoEmotions	N	48,837	43,410	5,427	28	1.18 ± 0.42
Amazon-531	Y	49,145	29,487	19,658	531	2.93 ± 0.26
DBPedia-298	Y	245,832	196,665	49,197	298	3.00 ± 0.00
NYT AC	Y	36,471	29,179	7,292	166	7.59 ± 5.61

BoW-based Methods. For the methods based on Bag of Words (BoW), we rely on the study by Galke et al. [45], who have evaluated various variants of an MLP operating on a bag-of-words. These include an MLP with one wide hidden layer and two hidden layers and different text representations as input, such as TF-IDF weighting, and using pre-trained word embeddings such as GloVe. We further list the numbers for fastText, SWEM, and logistic regression from Ding et al. [32]. Motivated by [175], we complement these numbers by running SVMs on unigram and trigram features.

Recurrent and Convolutional Neural Networks. We include scores of other sequence-based models such as LSTMs [32, 230] and CNNs [75, 227]. In contrast to the BoW-based models, these consider the sequence of the textual input and exploit this to train the classifier.

Small Language Models. We run own experiments with various pre-trained language models. The numbers reported in the literature for BERT applied to the same datasets differ a lot between some papers. Therefore, we fine-tune BERT and some of the most popular encoder-only language models, including DistilBERT, RoBERTa, DeBERTa, ALBERTv2, and ERNIE 2.0 ourselves. Since HBGL is the most promising hierarchy-aware approach using BERT, we also run own experiments with that model.

Large Language Models. We complement numbers from SLMs by results reported for LLMs. These include GPT-3.5 (`text-davinci-003`) with the CARP prompting strategy [158], a sophisticated few-shot learning technique that relies on sampling examples based on RoBERTa embeddings (R.S. variant) and voting among multiple runs. Note that in-context learning evaluations of LLMs, such as CARP, do not make use of the full training set but only few examples.

A different strategy we include here is to fine-tune the parameters of LLMs for text classification, as done in RGPT [228]. Specifically, RGPT consists of an ensemble of fine-tuned Llama-2 models [169].

Attention-free Language Models. Pre-trained attention-free models of attention-free architectures gMLP and aMLP are not available. Therefore, we train gMLP and aMLP from scratch.

Synthetic Text-Graph Methods. We consider both transductive as well as inductive graph-based methods for text classification. These include TextGCN along with its successors HeteGCN, TensorGCN, HyperGAT, DADGNN, Simplified GCN (SGC) [189], and many others. Ragesh et al. [139] evaluated a variant of TextGCN that is capable of inductive learning, which we include in our results.

Hierarchy-based Methods. For hierarchy-based text classification, we run experiments using HiAGM's best-performing variant, HiAGM-TP, with GCN as the graph encoder. We report the numbers of BERT+HiMatch and HGCLR. For comparison, we run several experiments with BoW-based methods and sequence-based methods that do not exploit the hierarchy but consider the classes as a set. For transformer-based models such as BERT, DeBERTa, and RoBERTa, we train them for up to 50 epochs with early stopping on Macro-F1 and patience of 5. In comparison to the multi-label datasets, longer training is needed for the hierarchical datasets.

6.3 Procedure

We distinguish between single-label and multi-label text classification settings. We apply standard train-test splits unless there is no default split provided (see Section 6.1).

For the single-label setting, we further distinguish between transductive and inductive text classification. In the transductive setting, as used in TextGCN and other synthetic text-graph approaches, the unlabeled test documents are visible during training. In the inductive setting, the test documents remain unseen until test time, i.e., they are not available for training or preprocessing. The distinction only matters for graph-based approaches because BoW-based and sequence-based models are usually inductive. We separately report the scores of the graph-based models for inductive and transductive setups from the literature, where available.

To avoid bias in the comparability of the results, we carefully checked all relevant parameters, such as the train/test splits, the number of classes in the datasets, whether datasets have been pre-processed to make the task substantially easier, and the evaluation metrics.

We repeat all of our own experiments five times with a different random initialization of the parameters and report the mean and standard deviation of these five runs. To tune the hyperparameters of the multi-label classification models, we choose randomly 20% of the train set as a validation set. Below, we provide a detailed description of the hyperparameter optimization and evaluation procedures for the models that we have run ourselves.

6.4 Hyperparameters for Own Experiments

We describe the choice and optimization of hyperparameters and relevant implementation details for single-label and multi-label classification. For each case, we follow the families of classification approaches.

Single-label Case. For the unigram and word-trigram SVM models, we first transform the features via TF-IDF and then employ a linear support vector classifier with hinge loss and default hyperparameters (l2 penalty, regularization strength $C = 1$).

For *fine-tuning the small language models* DistilBERT, RoBERTa, DeBERTA, ERNIE, ALBERT, and BERT-Large, we adopt the fine-tuning strategy of Galke and Scherp [45]. We fine-tune each model for 10 epochs with a linearly decaying learning rate. The initial learning rates are lr = $4.5 \cdot 10^{-5}$ for DistilBERT, $4 \cdot 10^{-5}$ for RoBERTa, $2 \cdot 10^{-5}$ for DeBERTA, $2.5 \cdot 10^{-5}$ for ERNIE, and $1 \cdot 10^{-5}$ for ALBERT and BERT-large. The batch sizes are 128 for DistilBERT, 16 for DeBERTA and BERT-large, and 32 for RoBERTa, ERNIE, and ALBERT. We truncate all inputs to 512 tokens. As common for BERT-like models, the sequence is pooled by taking the final representation of the first token and feeding it into an MLP module. We use the uncased versions of the pre-trained language models. For example, BERT-base refers to the "bert-base-uncased" model available on Hugging Face.[6]

Training gMLP/aMLP: We train the gMLP and aMLP models [97] from scratch on the text classification task and without any masked language model pre-training. There is an initial embedding layer, followed by 18 gMLP blocks with a token sequence length of 512. Layer normalization and

---

[6]https://huggingface.co/google-bert/bert-base-uncased

a GeLU activation function are applied between the blocks. For the aMLP version, we attach a single-head attention module to the spatial gating unit of size 64. We truncate all inputs to 512 tokens, use Adam optimizer with a learning rate of 10^{-4}, and run the training for 100 epochs with a batch size of 32. For pooling the sequence, we take the mean of the final layers' representations.

Multi-label Case. For *training the WideMLP* in the multi-label case, we tune the hyperparameters using a manual search on the R21578 dataset. We employ a TF-IDF input representation, 100 epochs, and a learning rate of 10^{-1} for all datasets. We increase the batch size according to the dataset size in order to limit the overall training time: For the smaller datasets (R21578, GoEmotions, Amazon-531, NYT AC, RCV1-V2), we use a batch size of 8. For DBPedia-298, the batch size is 32 and for EconBiz, we use a batch size of 256. The model is trained with binary cross-entropy.

At test time, class labels are assigned depending on a threshold on the class-specific output. It is common to threshold the sigmoidal output units at values of $\lambda = 0.5$ [171, 223]. However, Galke et al. [44] had found that a smaller threshold such as $\lambda = 0.2$ can be advantageous, especially in setups with an imbalanced label distribution. In pre-experiments with WideMLP, we tested different thresholds from $\lambda = 0.5$ to $\lambda = 0.1$ (0.1 steps) and experienced similar results reported in Galke et al. [44], where $\lambda = 0.2$ achieved the best results.

Fine-tuning BERT: For the BERT variant models, we use a manual search to find the best hyperparameters, and the same hyperparameters were chosen for all models. We fine-tune all parameters of the model, as we do for single-label classification. We use binary cross-entropy loss to reflect multi-label classification. We used the R21578 dataset for hyperparameter tuning and transferred the best hyperparameters to the other datasets. In practice, it has been observed that when using a larger batch, the quality of the model, as evaluated by its capacity to generalize, degrades [234]. In the pre-experiments with R21578, we found that small batch sizes are preferable for fine-tuning these models. We used a linearly decaying learning rate of $5 \cdot 10^{-5}$ with a batch size of 4 for all data sets. We truncate all inputs to 512 tokens. We fine-tune the models on the datasets for either 15 or 5 epochs for multi-label training. DBPedia-298 and GoEmotions had the best results with 5 epochs as the validation loss increases in subsequent iterations. For multi-label classification, we use a threshold of $\lambda = 0.5$ after pre-experiments with 0.2 and 0.5. As in the single-label case, we again use the uncased versions of the language models for our experiments. For HBGL, we use the hyperparameter values reported in the original work [66].

Training gMLP/aMLP: We use the same architecture as described in the single-label setup. Pre-experiments on the R21578 dataset have revealed 10^{-4} as the most suitable learning rate. The use of different learning-rate schedulers (linear decay, reduction on the plateau) was investigated, but we found the best results with a constant learning-rate schedule. We trained for 300 epochs with a batch size of 32 across all datasets except for Econbiz, where, due to the larger size of the dataset, we scale down our epoch count to 50 and set the batch size to 64. We collect results with a threshold of $\lambda = 0.2$ and $\lambda = 0.5$ and find that 0.5 leads to better results for gMLP and aMLP.

Training HiAGM: For hierarchical multi-label classification, we use HiAGM-TP, the best-performing variant of HiAGM with GCN as a structural encoder. We used the hyperparameters given in the original study [232]: a batch size of 64 with a learning rate of 10^{-4}. We train HiAGM for 300 epochs with early stopping based on validation loss with patience of 50 epochs. We experimented with a threshold of $\lambda = 0.5$ to $\lambda = 0.2$ in steps of 1/10 and found that 0.5 is preferable.

6.5 Measures

We report accuracy as the evaluation metric for single-label datasets. Note that the accuracy is equivalent to Micro-F1 in single-label classification [45]. We report the mean accuracy and standard deviation (SD) over five runs for neural network methods, which rely on random initialization

and other noise sources during training, such as dropout. For those models where we rely on numbers from the literature, we check if multiple runs are reported and include the corresponding information in our report. Note that the exact number of runs may differ from paper to paper. Most papers report five runs (if they have multiple runs), but others report ten runs.

For the multi-label datasets, we follow Galke et al. [44] and report the sample-based F1 measure. We chose this sample-based evaluation measure because it reflects the classification quality of each document separately. The sample-based F1 measure is calculated by the harmonic mean of precision and recall for each example individually, and then these scores are averaged. For comparability with scores reported in the literature, we also report the globally-averaged Micro-F1 and the class-averaged Macro-F1 for multi-label classification.

7 QUANTITATIVE COMPARISON

We present the results of our quantitative comparison, starting with the single-label datasets. Subsequently, we present our results of a sensitivity analysis of transformer models regarding the fine-tuning learning rate to explain the differences in performance found in the literature. This is followed by the results on multi-label and hierarchical text classification. Finally, we report the parameter counts of selected models.

7.1 Single-label Text Classification

Table 3 shows the results of the inductive single-label text classification on the five datasets, while the results of the transductive methods are reported in Table 4. Regarding the inductive text classification, one sees that sequence-based transformers are overall the best methods. The sequence-based transformer DeBERTa attains the highest scores. The margin to a standard BERT model is most notable on ohsumed (75.9 vs. 71.5) and on MR (90.0 vs. 86.6).

The family of graph-based methods shows good performance but is about one point behind, for some datasets even more. The BoW-based methods overall achieve strong performance, up to a point where a BoW-based WideMLP matches or even outperforms the graph-based methods in the inductive setting. In the transductive setting shown in Table 4, the graph-based methods can use unlabeled test data and increase their scores. Within the transductive setting, all graph-based methods achieve quite similar accuracy results. In the inductive case, the difference between the graph-based methods and the other families is much higher.

We describe the results reported in Table 3 regarding the accuracy scores in the inductive setting in more detail. In the inductive setting, the WideMLP models perform best among the BoW-based methods, in particular, TF-IDF+WideMLP and WideMLP on an unweighted BoW. Another observation is that an MLP with one hidden layer (but wide) is sufficient for our considered datasets. The scores for the MLP variants with 2 hidden layers (WideMLP-2) are consistently lower. We further observe that pure BoW and TF-IDF-weighted BoW yield better results than approaches that exploit pre-trained word embeddings such as GloVe-MLP, fastText, and SWEM.

Also, SVMs and logistic regression are strong text classification methods. When modifying the TF-IDF-weighting to incorporate weights from matching text tokens to the (descriptive) names of the classes, one observes that results improve further. For example, the CFE-IterativeAdditive method uses a linear SVM with term-based substring matching (from the documents) to the class names [2]. It uses this label matching of the terms to adapt the global IDF weights iteratively, denoted as TF-ICF.

The best-performing graph-based model not using a pre-trained language model is TextSSL, closely followed by HyperGAT. Only with the help of a pre-trained language model, ConTextING-RoBERTa attains higher scores on R8, R52, ohsumed,and MR. The largest difference is found on the MR sentiment analysis dataset, where ConTextING-RoBERTa reaches 89.43 compared to 77.08 of

Table 3. Results for the inductive training on the single-label text classification datasets. For our experiments, we report the mean accuracy and standard deviation (SD) over five runs. For numbers from the literature, we report the SD if available. GPT-3 methods use 16 examples per class for in-context learning, and the (R.S)-variant uses a RoBERTa sampler to select these examples. Column "Provenance" reports the source. d. f. is short for the use of a different variant of the dataset or a different split, and thus, the number is omitted to ensure comparability.

Inductive Setting	20ng	R8	R52	ohsumed	MR	Provenance
BoW-based Methods						
Logistic regression with TF-IDF	83.70	93.33	90.65	61.14	76.28	[139]
Unigram SVM with TF-IDF	83.44	97.49	94.70	67.40	76.36	our experiment
Trigram SVM with TF-IDF	83.39	97.21	93.85	69.30	77.35	our experiment
XGBoost with TF-IDF	73.29	94.75	88.82	58.27	64.46	our experiment
WideMLP with TF-IDF	$84.20_{0.16}$	$97.08_{0.16}$	$93.67_{0.23}$	$66.06_{0.29}$	$76.32_{0.17}$	[45]
WideMLP	$83.31_{0.22}$	$97.27_{0.12}$	$93.89_{0.16}$	$63.95_{0.13}$	$76.72_{0.26}$	[45]
WideMLP-2	$81.02_{0.23}$	$96.61_{1.22}$	$93.98_{0.23}$	$61.71_{0.33}$	$75.91_{0.51}$	[45]
GloVe+WideMLP	$76.80_{0.11}$	$96.44_{0.08}$	$93.58_{0.06}$	$61.36_{0.22}$	$75.96_{0.17}$	[45]
GloVe+WideMLP-2	$76.33_{0.18}$	$96.50_{0.14}$	$93.19_{0.11}$	$61.65_{0.27}$	$75.72_{0.45}$	[45]
SWEM	$85.16_{0.29}$	$95.32_{0.26}$	$92.94_{0.24}$	$63.12_{0.55}$	$76.65_{0.63}$	[32]
fastText	$79.38_{0.30}$	$96.13_{0.21}$	$92.81_{0.09}$	$57.70_{0.49}$	$75.14_{0.20}$	[32]
CFE-IterativeAdditive	$85.51_{0.04}$	$97.94_{0.02}$	$95.13_{0.04}$	$68.90_{0.02}$	—	[2]
Sequence-based Methods						
CNN+GloVe	82.15	95.71	87.59	58.44	77.75	[60]
CNN-non-static	—	—	—	—	81.5	[75]
Word2Vec+CNN	—	—	—	—	81.24	[227]
GloVe+CNN	—	—	—	—	81.03	[227]
LSTM w/ pre-training	$75.43_{1.72}$	$96.09_{0.19}$	$90.48_{0.86}$	$51.10_{1.50}$	$77.33_{0.89}$	[32]
Bi-LSTM (GloVe)	—	96.31	90.54	—	77.68	[230]
GPT-3.5 full finetuning via OpenAI API	—	—	$95.27_{0.55}$	$51.84_{0.45}$	—	[86]
Bloom-7.1B (4-bit+LoRA) full finetuning	—	—	—	$67.54_{0.6}$	—	[86]
Llama2-7B (4-bit+LoRA) full finetuning	—	—	—	$67.66_{0.72}$	—	[86]
Llama3-8B (4-bit+LoRA) full finetuning	—	—	—	$68.02_{0.26}$	—	[86]
GPT-3.5 16-shot	—	91.58	91.56	—	89.15	[158]
GPT-3.5+CoT 16-shot	—	92.49	92.03	—	89.91	[158]
GPT-3.5+CARP 16-shot	—	97.60	96.19	—	90.03	[158]
GPT-3.5 (R.S.) 16-shot	—	95.57	95.79	—	90.90	[158]
GPT-3.5+CoT (R.S.) 16-shot	—	95.59	95.89	—	90.17	[158]
GPT-3.5+CARP+vote (R.S.) 16-shot	—	98.78	96.95	—	92.39	[158]
RGPT "Pushing the Limit"	—	—	—	77.41	—	[228]
QLFR 20-shot	—	—	—	61.10	81.70	[190]
LLMEmbed bert+roberta+llama embeddings	—	98.22	95.68	—	d. f.	[96]
BERT-base	$87.21_{0.18}$	$98.03_{0.24}$	$96.17_{0.33}$	$71.46_{0.54}$	$86.61_{0.38}$	[45]
AM-BERT	89.03	98.43	97.17	73.47	86.83	[212]
AM-RoBERTa	90.32	98.97	98.12	73.89	89.75	[212]
BERT-large	$85.83_{0.64}$	$97.98_{0.29}$	$96.41_{0.28}$	$72.69_{0.63}$	$88.22_{0.21}$	our experiment
DistilBERT	$86.90_{0.04}$	$97.93_{0.11}$	$96.89_{0.12}$	$71.65_{0.38}$	$85.11_{0.25}$	our experiment
RoBERTa	$86.80_{0.51}$	$98.19_{0.18}$	$97.13_{0.10}$	$75.08_{0.42}$	$88.68_{0.29}$	our experiment
DeBERTa	$87.60_{0.45}$	$98.30_{0.20}$	$97.10_{0.13}$	$75.94_{0.33}$	$89.98_{0.26}$	our experiment
ERNIE 2.0	$87.79_{0.29}$	$97.95_{0.16}$	$96.96_{0.23}$	$73.33_{0.30}$	$89.19_{0.24}$	our experiment
ALBERTv2	$82.08_{0.30}$	$97.88_{0.22}$	$94.95_{0.20}$	$62.31_{2.11}$	$86.28_{0.21}$	our experiment
gMLP w/o pre-training	$68.62_{1.66}$	$94.46_{0.41}$	$91.27_{0.99}$	$39.58_{0.77}$	$66.24_{0.37}$	our experiment
aMLP w/o pre-training	$72.14_{1.07}$	$95.40_{0.20}$	$91.77_{0.11}$	$49.29_{1.13}$	$66.67_{0.35}$	our experiment
BERT w. token-level GCN New!	80.12	97.62	94.09	65.98	82.57	[34]
LFTC (compression w. 1-NN) New!	81.4	96.5	90.6	43.5	—	[110]

Graph-based Methods						
Text-level GNN	—	$97.8_{0.2}$	$94.6_{0.3}$	$69.4_{0.6}$	—	[59]
TextING-M	—	$98.13_{0.31}$	$95.68_{0.35}$	$70.84_{0.52}$	$80.19_{0.31}$	[229]
TextGCN	$80.88_{0.54}$	$94.00_{0.40}$	$89.39_{0.38}$	$56.32_{1.36}$	$74.60_{0.43}$	[139]
HeteGCN	$84.59_{0.14}$	$97.17_{0.33}$	$93.89_{0.45}$	$63.79_{0.80}$	$75.62_{0.26}$	[139]
HyperGAT-ind	84.63	97.03	94.55	67.33	$77.08_{0.27}$	[60]
DADGNN	—	$98.15_{0.16}$	$95.16_{0.22}$	—	$78.64_{0.29}$	[104]
SGNN	—	98.09	95.46	—	80.58	[230]
ESGNN	—	98.23	95.72	—	80.93	[230]
C-BERT (ESGNN+BERT)	—	98.28	96.52	—	86.06	[230]
ConTextING-RoBERTa	85.00	98.13	96.40	72.53	89.43	[60]
TextSSL	$85.26_{0.28}$	$97.81_{0.14}$	$95.48_{0.26}$	$70.59_{0.38}$	$79.74_{0.19}$	[134]
GLTC	—	98.17	95.77	71.82	80.29	[203]
InducT-GCN	$84.03_{0.06}$	$96.64_{0.03}$	$93.16_{0.13}$	$65.87_{0.16}$	$75.21_{0.08}$	our experiment
MHGAT	$92.68_{0.30}$	$97.65_{0.47}$	$94.78_{0.37}$	$72.88_{0.84}$	$78.09_{0.73}$	[68]

Table 4. Results for the single-label text classification datasets. Note that only graph-based methods require the transductive setting. We report mean accuracy and standard deviation over five runs. The column "Provenance" reports the source.

Transductive Setting	20ng	R8	R52	ohsumed	MR	Provenance
Graph-based Methods						
TextGCN	86.34	97.07	93.56	68.36	76.74	[200]
SGC	$88.5_{0.1}$	$97.2_{0.1}$	$94.0_{0.2}$	$68.5_{0.3}$	$75.9_{0.3}$	[189]
TensorGCN	87.74	98.04	95.05	70.11	77.91	[103]
HeteGCN	$87.15_{0.15}$	$97.24_{0.51}$	$94.35_{0.25}$	$68.11_{0.70}$	$76.71_{0.33}$	[139]
HyperGAT	$86.62_{0.16}$	$97.07_{0.23}$	$94.98_{0.27}$	$69.90_{0.34}$	$78.32_{0.27}$	[32]
BertGCN	89.3	98.1	96.6	72.8	86.0	[94]
RoBERTaGCN	89.5	98.2	96.1	72.8	89.7	[94]
TextGCN-BERT-serial-SB	—	97.78	94.08	68.83	86.69	[216]
TextGCN-CNN-serial-SB	—	$98.53_{0.21}$	$96.35_{0.09}$	$71.85_{0.49}$	$87.59_{0.20}$	[216]
AGGNN	—	$98.18_{0.10}$	$94.72_{0.29}$	$70.26_{0.38}$	$80.03_{0.22}$	[29]
STGCN	—	97.2	—	—	78.2	[201]
STGCN+BERT+BiLSTM	—	98.5	—	—	82.5	[201]
CTGCN	86.92	97.85	94.63	69.73	77.69	[194]
TSW-GNN	—	$97.84_{0.4}$	$95.25_{0.1}$	$71.36_{0.3}$	$80.26_{0.6}$	[89]
KGAT	—	97.41	95.00	70.24	79.03	[180]
IMGCN	—	98.34	—	—	87.81	[195]
BERT+SGC	—	98.04	96.34	—	86.63	[88]

HyperGAT-ind. It should be noted that the difference of the graph-based ConTextING-RoBERTa to a plain RoBERTa-base model on MR is less than one point. Furthermore, a BoW-based logistic regression outperforms the graph-based TextGCN on four out of five benchmark datasets.

The sequential MLP-based models gMLP and aMLP show poor performance in our experiments without pre-training. Including single-head attention layers in aMLP increased accuracy scores by 0.5 to 10 points compared to the gMLP. The overall performance of aMLP is still much lower than BERT and does not exceed a simple logistic regression on three of five data sets.

In summary, fine-tuned transformers yield the highest scores. DistilBERT outperforms the best pure graph-based method HyperGAT by 7 points on the MR dataset while being on-par on the others. Comparing DeBERTa with the best graph-based method ConTextING-RoBERTa, there is still superiority of the pure transformer, but the margin is smaller. Regarding BERT-large, we observe that the scores are improved over BERT-base by a small 1 point for the ohsumed and MR datasets,

but the inverse of a performance decrease of 1 point is recorded for 20ng. For R8 and R52, both BERT-base and BERT-large achieve about the same performance.

The use of GPT-3 in a 16-shot setting in CARP [158] does not reach the performance of the encoder-only language models. The results can be improved by adding dedicated prompting strategies and non-uniform samplers. The increase is particularly notable on R8 and R52. With these prompting strategies and 16 examples per class in the prompt, GPT-3 performs barely below the encoder-only language models on R8 and R52 but yields the overall best results for the sentiment classification task MR.

7.2 Sensitivity to Fine-tuning Learning Rate

While analyzing the numbers reported in the papers, we noticed that the performance of BERT and other transformer models differs from paper to paper. To shed light on the differences between BERT results on the same datasets, we repeat experiments with different, most importantly, lower learning rates during fine-tuning. The results are shown in Table 5. We observe that there are substantial differences between the supposedly same BERT models reported in the literature. For BERT-base, the difference is in many cases 2 and 3 points on the 20ng and ohsumed datasets, respectively. For RoBERTa, we even observe deviations of more than 3 points on 20ng and 5 points on ohsumed despite using the same learning rate.

Some of the reported numbers for fine-tuned BERT models are even far behind the others. For example, Yin et al. [203] report BERT-base results that are more than ten points behind the others on MR and ohsumed. We hypothesize that this discrepancy is caused by a suboptimal choice of hyperparameters, e. g., a too-high learning rate, which are unfortunately not provided.

On the R8, R52, and MR datasets, the results differ by not more than 1 point. Remarkably, the lightweight DistilBERT is quite sensitive to a small change in the learning rate. For example, the difference of more than 1 point on R52 and even 2 points on ohsumed is caused by changing the learning rate by a factor of only $0.5 \cdot 10^{-5}$.

Table 5. Comparison of different transformer models and hyperparameter settings. We report mean accuracy and standard deviation over five runs on the single-label text classification datasets (inductive). Column "Provenance" reports the source. N/P refers to the case where the paper (or potential supplementary materials) did not provide information about the learning rate.

Inductive Setting	20ng	R8	R52	ohsumed	MR	Provenance
BERT-base (lr = $5 \cdot 10^{-5}$)	$87.21_{0.18}$	$98.03_{0.24}$	$96.17_{0.33}$	$71.46_{0.54}$	$86.61_{0.38}$	our experiment
BERT-base (lr = $5 \cdot 10^{-5}$)	—	—	$95.55_{0.36}$	$65.71_{0.30}$	—	[86]
BERT-base (lr = $3.5 \cdot 10^{-5}$)	$87.31_{0.21}$	$98.19_{0.12}$	$97.13_{0.16}$	$73.54_{0.45}$	$86.86_{0.10}$	our experiment
BERT-base (lr = $2 \cdot 10^{-5}$)[1]	85.20	97.73	96.22	70.53	85.71	[212]
BERT-base (lr = $1 \cdot 10^{-5}$)	84.54	97.26	96.26	68.74	85.88	[60]
BERT-base (lr = $1 \cdot 10^{-5}$)	85.3	97.8	96.4	70.5	85.7	[94]
BERT-base (lr = N/P)[1]	—	96.78	91.35	60.46	76.13	[203]
BERT-base (lr = N/P)	—	98.2	—	—	85.7	[201]
BERT-base (lr = N/P)	83.1	—	—	—	—	[18]
DistilBERT (lr = $5 \cdot 10^{-5}$)	$86.24_{0.26}$	$97.89_{0.15}$	$95.34_{0.08}$	$69.08_{0.60}$	$85.10_{0.33}$	our experiment
DistilBERT (lr = $4.5 \cdot 10^{-5}$)	$86.90_{0.04}$	$97.93_{0.11}$	$96.89_{0.12}$	$71.65_{0.38}$	$85.11_{0.25}$	our experiment
RoBERTa-base (lr = $4 \cdot 10^{-5}$)	$86.80_{0.51}$	$98.19_{0.18}$	$97.13_{0.10}$	$75.08_{0.42}$	$88.68_{0.29}$	our experiment
RoBERTa-base (lr = $4 \cdot 10^{-5}$)	83.8	97.8	96.2	70.7	89.4	[94]
RoBERTa-base (lr = $1 \cdot 10^{-5}$)	84.07	97.35	95.48	69.86	87.08	[60]
RoBERTa-base (lr = N/P)[1]	83.80	97.80	96.20	70.70	89.40	[212]

[1] Authors applied special preprocessing of the input text.

7.3 Multi-label Text Classification

Table 6. Results for the inductive multi-label text classification datasets. We report the sample-based F1 metric to reflect how well the classifier performs on average per a set of new documents. An "NA" indicates that HiAGM could not be applied to the dataset since the classes are not hierarchically organized. "OOM" denotes that the model ran out of memory. Standard deviation across runs is denoted in braces.

Inductive Setting	R21578	RCV1-V2	EconBiz	Amaz.	DBPedia	NYT	GoEmo.
BoW-based methods							
WideMLP	80.41	69.92 $_{0.11}$	23.15	59.92	89.47	62.38 $_{0.27}$	37.13
TF-IDF WideMLP	88.15	81.51 $_{0.03}$	45.38	80.32	94.91	75.58 $_{0.09}$	40.07
Sequence-based methods							
BERT-base	92.21	88.16 $_{0.16}$	42.08	86.69	97.66	79.11 $_{0.22}$	54.18
BERT-large	92.23	88.83 $_{0.17}$	33.62	88.34	97.69	80.32 $_{0.39}$	54.02
DistilBERT	92.11	87.50 $_{0.11}$	39.41	87.47	97.58	79.18 $_{0.17}$	55.95
RoBERTa	90.85	88.62 $_{0.21}$	40.56	86.21	97.26	79.14 $_{0.55}$	54.64
DeBERTa	91.24	88.45 $_{0.19}$	41.43	89.21	97.65	79.95 $_{0.40}$	56.51
HBGL	-	88.76 $_{0.24}$	-	-	-	82.01 $_{0.22}$	-
gMLP w/o pre-train	85.39	79.11	40.53	83.72	95.07	72.23	44.92
aMLP w/o pre-train	85.76	77.87	42.11	82.33	95.79	70.88	47.19
Hierarchy-based methods							
HiAGM-TP+GCN	—	85.51 $_{0.11}$	OOM	89.05	97.17	76.57 $_{0.17}$	—

Table 6 shows the sample-based F1 results of the multi-label text classification methods. Overall, the sequence-based models perform best, except for the Econbiz dataset. The best-performing models depend on the datasets. For some, like DBpedia, the difference between the follow-up models is very small, while for others, a difference of up to two points can be observed between the transformers. HBGL is the best model on the NYT dataset, with about 2 points better than DeBERTa and the other transformers. DeBERTa and the other transformers are on par with HBGL on the RCV1-V2 dataset. Regarding BERT-large one notices that for five out of the seven datasets, the results are marginally better than BERT-base. Only for Amazon, BERT-large improves the results by more than one point. However, BERT-large only obtains a sample-based F1 score of 33.62 on EconBiz, compared to 42.08 achieved by BERT-base. The hierarchy-based method HiAGM-TP+GCN overall shows strong performance. It is on par with the transformers on Amazon and DBPedia and about 3 points behind the best transformer on RCV1-V2 and NYT. The method ran out of memory (OOM) on the EconBiz dataset with the largest number of classes.

Comparing the MLP-based methods, the WideMLP is better than the sequential MLP-based models on R21578, RCV1-V2, EconBiz, and NYT, on par with DBPedia-298, and only falling behind gMLP and aMLP on Amazon-531 and GoEmotions. The sequence-based aMLP is on par with BERT on EconBiz. On the sentiment prediction task in GoEmotions, the WideMLP performs worst. However, the TF-IDF+WideMLP outperforms the pre-trained transformers on EconBiz. The improvement over the best transformer is more than 3 points.

7.4 Hierarchical Text Classification

For the multi-label datasets, we reported the sample-based F1 score in Table 6. We argue that the sample-based F1 represents real-world applications where each document needs to be annotated one document after the other such as in subject indexing by librarians [44, 109]. Since the literature on hierarchical multi-label classification frequently reports Micro-F1 and Macro-F1 scores, we also

Table 7. Mean accuracy and standard deviation (where available) across five runs for hierarchical multi-label classification on three common benchmark datasets using Micro-F1 and Macro-F1 scores.

Model	WOS (Micro/Macro)	NYT (Micro/Macro)	RCV1-V2 (Micro/Macro)	Provenance
BoW-based methods				
WideMLP	—	$57.18_{0.28}$ / $21.96_{0.19}$	$68.31_{0.12}$ / $27.88_{0.49}$	our experiment
TF-IDF WideMLP	—	$74.53_{0.07}$ / $56.11_{0.16}$	$80.45_{0.02}$ / $53.27_{0.09}$	our experiment
Sequence-based methods				
BERT-base	$86.19_{0.11}$ / $80.23_{0.20}$	$79.07_{0.22}$ / $67.63_{0.42}$	$86.38_{0.23}$ / $67.89_{1.42}$	our experiment
BERT-base	85.63 / 79.07	78.24 / 65.62	85.65 / 67.02	[183]
BERT-base	86.26 / 80.58	—	86.26 / 67.35	[20]
BERT-large	$86.66_{0.31}$ / $80.80_{0.38}$	$80.69_{0.08}$ / $70.27_{0.39}$	$87.40_{0.11}$ / $70.15_{0.39}$	our experiment
DeBERTa-base	$86.82_{0.11}$ / $80.85_{0.48}$	$81.21_{0.17}$ / $71.33_{0.26}$	$87.24_{0.18}$ / $69.66_{0.52}$	our experiment
DeBERTaV3-base	$86.58_{0.24}$ / $80.41_{0.69}$	$79.96_{0.20}$ / $67.30_{0.23}$	$86.77_{0.18}$ / $67.49_{1.87}$	our experiment
RoBERTa-base	$86.35_{0.15}$ / $80.16_{0.13}$	$81.47_{0.15}$ / $71.45_{0.45}$	$87.38_{0.17}$ / $68.60_{1.40}$	our experiment
ModernBERT-base New!	$86.15_{0.33}$ / $80.34_{0.22}$	$79.83_{0.53}$ / $68.99_{0.54}$	$86.20_{0.42}$ / $67.43_{0.66}$	our experiment
BART	84.08 / 77.43	19.21 / 6.49	86.20 / 65.11	[206]
T5	82.03 / 74.62	46.71 / 20.06	84.9 / 57.01	[206]
SGM	67.74 / 74.01	64.68 / 72.78	71.85 / 35.29	[206]
SGM-T5	85.83 / 80.79	—	84.39 / 65.09	[208]
Seq2Tree	87.20 / 82.50	—	86.88 / 70.01	[208]
RADAr	$87.17_{0.04}$ / $81.84_{0.08}$	$79.84_{0.07}$ / $68.64_{0.28}$	$87.23_{0.05}$ / $69.64_{0.12}$	[206]
HBGL	87.36 / 82.00	80.47 / 70.19	87.23 / 71.07	[66]
HBGL	87.68 / 82.01	$80.01_{0.22}$ / $70.14_{0.27}$	$86.94_{0.26}$ / $70.49_{0.58}$	our experiment
Retrieval-style ICL $_{16\text{-shot}}$ New!	$81.12_{0.26}$ / $73.72_{0.17}$	—	—	[21]
Hierarchy-based methods				
BERT+HiMatch	86.70 / 81.06	—	86.33 / 68.66	[20]
HiAGM-TP+GCN	85.82 / 80.28	74.97 / 60.83	83.96 / 63.35	[232][232]
HiAGM-TP+GCN	—	$74.73_{0.08}$ / $58.44_{0.25}$	$83.95_{0.11}$ / $62.13_{0.35}$	our experiment
HGCLR	87.11 / 81.20	78.86 / 67.96	86.49 / 68.31	[183]
HGBL New!	87.07 / 81.10	78.55 / 67.08	87.55 / 68.10	[217]
HALB	87.45 / 82.04	79.56 / 69.28	86.94 / 69.32	[220]
HILL	87.28 / 81.77	80.47 / 69.96	87.31 / 70.12	[237]
HE-AGCRCNN	—	—	77.8 / 51.3	[130]
K-HTC New!	87.29 / 81.69	—	—	[106]

report them in Table 7. Here, we use common benchmark datasets for hierarchical text classification. These are Web of Science (WoS) [77], NYT, and RCV1-v2.

We can again see that sequence-based models perform better than the hierarchy-based methods. The best method is HBGL, with between 2 and 3 points advantage in Micro-F1 and 1 to 2 points in Macro-F1 over the strongest graph-based competitor HGCLR. Interestingly, HBGL scores 3 to 5 points higher than a pure BERT model. The Micro-F1 results for BERT on the NYT and RCV1-V2 datasets by Wang et al. [183], Chen et al. [20], and own experiments, are very similar. It is notable that for the Macro-F1 scores, our experiments show a drop of about 7 points compared to the literature such as [20, 183]. One difference in these experiments is that we use a learning rate of $\text{lr} = 5 \cdot 10^{-5}$, while Wang et al. [183] use $\text{lr} = 3 \cdot 10^{-5}$ and Chen et al. [20] apply BERT $\text{lr} = 2 \cdot 10^{-5}$.

7.5 Parameter Count of Models

Table 8 lists the parameter counts of selected methods used in our experiments. The parameter counts are the same for the multi-label and single-label setups except for a small variation depending

on the number of classes. Even though the MLP is fully connected on top of a bag of words with the dimensionality of the vocabulary size, it has only half of the parameters as DistilBERT and a quarter of the parameters of BERT-base. Using TF-IDF does not change the number of model parameters. The MLP-based models gMLP and aMLP are larger than the WideMLP models but still less than half the size of BERT-base. Due to the high vocabulary size, GloVe-based models have many parameters, but most parameters are frozen, i. e., not updated during training. HiAGM has about as many parameters as gMLP and aMLP, less than DistilBERT, and half as many as BERT-base. BERT-large has about three times the number of parameters than BERT-base. RoBERTa-base and DeBERTa-base have more parameters than BERT-base but fall in the same order of magnitude. HBGL essentially uses a BERT model, which results in 110M parameters.

Table 8. Parameter counts for selected methods used in our comparison

Model	#parameters
BoW-based methods	
TF-IDF WideMLP	31.3M
WideMLP	31.3M
WideMLP-2	32.3M
GloVe+WideMLP	575,2M (frozen) + 0.3M
GloVe+WideMLP-2	575,2M (frozen) + 1.3M
Sequence-based methods	
BERT-base	110M
BERT-large	336M
DistilBERT	66M
RoBERTA	123M
DeBERTA	134M
ERNIE-base 2.0	110M
ALBERTv2	12M
gMLP	48.5M
aMLP	51.4M
HBGL	110M
GPT-3	175B
Graph/Hierarchy-based methods	
HyperGAT	LDA parameters + 3.1M
HiAGM	53.9M
ConTextING-RoBERTa	129M

8 DISCUSSION

8.1 Fine-tuned SLMs Preferable over In-context Learned LLMs

The state of the art in single-label text classification is held by fine-tuned language models. More specifically, DeBERTa has a slight edge over RoBERTa and BERT. Surprisingly, BERT-large does not improve more than 1 point on the single-label datasets compared to BERT-base, despite having three times more parameters. On 20ng, the performance even drops by one point. Presumably, the capacity of the BERT-base is already sufficient to tackle the single-label classification tasks, especially for the R8 and R52 datasets. At the same time, BERT-large is known to have difficulties in fine-tuning on smaller datasets [31].

Our analysis shows that graphs synthesized from the text provide little to no additional value in graph-based methods. Even traditional methods like an SVM based on word tri-grams outperform many recently proposed methods based on graph neural networks on single-label datasets.

For hierarchical multi-label text classification, we come to a similar conclusion. There are tremendous efforts to incorporate graph neural networks, e.g., to use a GNN to encode the class hierarchy, as in HiAGM. However, the best-performing model is HBGL, which leverages BERT to make use of the label hierarchy. Various other methods, including mixtures of BERT and GNNs, fail to outperform the best of our tested language models.

What matters for model performance is the distinction between in-context learning and fine-tuning. Generally, fine-tuning on the full training set yields better results. While SLM *need* fine-tuning to obtain their good results, LLM can also do in-context learning with few examples, but it is by far not as good as fine-tuning. Thus, it is no surprise when fine-tuned LLMs yield (slightly) better scores than fine-tuned SLMs at the expense of a few billion trainable parameters. For instance, fine-tuned Llama-2 with 7 billion parameters hardly outperforms a BERT model with a mere 100M parameters. Other methods, such as CARP, rely on fine-tuning an SLM and using an SLM to select examples for the LLM. While analyzing the importance of example selection is of scientific interest, practitioners should take into account that one could use the fine-tuned SLM model directly for classification and get better results than the combination of a fine-tuned SLM and in-context learning with an LLM.

We expect similar observations to be made on other text classification datasets because we have already covered a wide range of text classification settings: long, medium, and short texts, topic and sentiment classification, single-label and multi-label, and hierarchical classification in the domains of forum postings, news, movie reviews, scholarly articles, and product reviews. A recent study by Edwards and Camacho-Collados [38] confirms the finding that smaller models fine-tuned on the full training set outperform few-shot-prompted larger models. Bucher et al. [14] and Lepagnol et al. [83] report similar findings. Yehudai and Bendel [202] show that even in few-shot scenarios, a fine-tuned SLM yields better performance than generative language models with in-context learning, supporting our main finding that fine-tuning is what matters for text classification. The strength of bag-of-words methods [45] has further been replicated [124][7] and confirmed by other studies [224].

Finally, it is worth noting that LFTC is an approach based on data compression and 1-nearest neighbor (1-NN) [110]. It sticks out insofar that it does not require learning but still can be considered a sequence-based model. LFTC shows strong performance on the 20ng, R8, and R52 datasets but the worst performance of all models on the ohsumed dataset.

8.2 Subpar Language Model Performance Can Be Pushed via Prompting Schemes, Ensembling, and Fine-tuning

Large language models such as GPT-3 can be employed for classification via in-context learning. If the language model is prompted without a single example in the prompt, it relies on the name of the class being descriptive [147]. If the name of the class is not descriptive (e.g., an identifier), then it is necessary to provide a few examples per class in the prompt. Notably, this in-context learning strategy yields reasonable performance while requiring only a few labeled examples [158, 228]. Still, the final performance is substantially lower than fully fine-tuned encoder-only small language models.

The best prompting technique using CARP [158], i.e., the variant of GPT-3+CARP+vote (16-shot, RoBERTA sampler) requires a fine-tuned RoBERTa model for sampling the 16 most representative training examples. This implies the need for a full fine-tuning of a transformer model on the corpus

---

[7]https://github.com/SahanaRamnath/bow-vs-graph-vs-seq-textclassification

prior to prompting the decoder-only language model. Furthermore, it should be taken into account that datasets with long documents and/or a large number of classes can lead to exceeding the context window of the language model, which is a limitation of this approach but also provides opportunities for future research.

In line with our claim that the important distinction is fine-tuning vs. in-context learning, approaches that incorporate fine-tuning produce better results than LLMs applied with in-context learning [92, 228]. A key trick seems to be removing the constraint of the left-to-right attention mask. Zhang et al. [228] instead trains an ensemble of fine-tuned Llama models to surpass the performance of BERT/RoBERTa. For practical applications, it is worth considering that the compared Llama models have 8B parameters (multiplied by ensemble size), while SLMs have only about 100M parameters.

8.3 Synthetic Text-graphs Hardly Bring an Advantage

Interestingly, our experiments show that BoW-based models like WideMLP and SVM outperform the recent graph-based models TextGCN, HeteGCN, and Induct-GCN in the inductive text classification setting. One exception is 20ng, where Induct-GCN outperforms the SVM models. Trigram SVM is the best BoW-based model for ohsumed. Notably, the use of concept-based TF-ICF features in CFE-IterativeAdditive [2] improves the result in three datasets. A similar observation was made by Galke et al. [44] who used CTF-IDF features, i. e., extracted concepts defined in the label hierarchy, reweighted them by IDF, and concatenated them with a standard TF-IDF vector. For this CTF-IDF representation, the term frequencies are supplemented by concept frequencies based on an exact string matching to the concept labels, as it is also done by Attieh and Tekli [2].

On four datasets, including the RCV1-V2 and NYT benchmarks, Galke et al. observed a consistent improvement in using concept-based features in addition to term-based features and only using concept-based features, respectively. The strong performance of pure BoW-MLP questions the added value of synthetic graphs in models like TextGCN and Induct-GCN to the topical text classification task. Therefore, we argue that using strong baseline models for text classification is important to assess true scientific progress [26].

Graph-based methods come with high training costs. First, the graph has to be computed. Second, a GNN has to be trained. For standard GNN methods, the whole graph has to fit into the GPU memory, and mini-batching is non-trivial but possible with dedicated sampling techniques for GNNs [40]. Notably, none of the recent works on text classification have employed such dedicated sampling techniques. Note that word-document graphs require $O(N^2)$ space, where N is the number of documents plus the vocabulary size, which is a hurdle for large-scale applications.

In the transductive setting, graph-based text classification models show a large margin over an MLP. However, as argued in the introduction, transductive models have the strong drawback of being unable to apply to documents not seen during training. The only application scenario for transductive models is where a partially labeled corpus should be fully annotated. Follow-up approaches such as TensorGCN also suffer from these limitations. However, recent extensions such as HeteGCN, HyperGAT, InductGCN, HieGAT, and DADGNN already relaxes this constraint and enables inductive learning. But as argued above, these inductive graph-based models fail to outperform even simple baselines like an MLP or an SVM.

According to the data processing inequality [25], transforming a text corpus into a graph cannot add any new information. The seminal paper on graph convolution et al. [76] argued that graph neural networks are most effective when the edges provide additional information that cannot be modeled otherwise. Therefore, it is important to distinguish between text-induced graphs for text classification, which seem to provide little to no gain, and tasks where the *natural* structure of the graph data provides more information than the mere text, e. g., citation networks. When extra

information is encoded in the graph, graph neural networks are the state of the art [76, 174] and superior to MLPs that use only the node features and not the graph structure [148]. However, our work suggests that a graph induced from pure text does not provide such additional information and thus does *not* improve text classification results over the state of the art. Recently, Bugueno and de Melo [15] compared different document representations (Word2vec, GloVe, and frozen BERT) for graph neural networks. Using different datasets, they confirm that on most datasets, graph neural networks did *not* outperform a fine-tuned BERT regardless of the choice of input representation. In addition, the finding can be also confirmed in a study on short text classification [72], where several graph-based methods have been compared to SLMs on six benchmark datasets of short text (including R8 and MR) and four new datasets.

Despite all recently proposed approaches to text classification, fine-tuning a pre-trained language model remains the state of the art. Text-induced graph-based methods only marginally improve the classification accuracy in comparison to bag-of-words models.

8.4 Using a Graph Encoder for the Hierarchy Hardly Brings an Advantage for Hierarchical Text Classification

In multi-label classification, we make similar observations as in the single-label case. Encoder-only models like DeBERTa and RoBERTa, the HBGL method, which incorporates the hierarchy into a standard BERT model, and RADAr, which uses an autoregressive decoder instead of a classifier head, are the overall best-performing models depending on the dataset and metric. HiAGM uses a GNN to encode the class hierarchy but fails to outperform the hierarchy-agnostic sequence-based DeBERTa model. In general, WideMLP is a strong baseline in the multi-label setup, like in single-label text classification. It is achieving performance comparable to that of the transformers and HiAGM. Notably, the bag-of-words WideMLP is the strongest method for the largest dataset, EconBiz, with thousands of classes. This may be due to the highly imbalanced (long tail) label distribution of the EconBiz dataset [109], which may be easier to reflect in a model trained from scratch than in a pre-trained model, such as BERT.

HiAGM's performance is comparable to that of DistilBERT and BERT. However, HiAGM cannot be used with the R21578 and GoEmotions datasets because they do not have label hierarchies. Additionally, large hierarchies, such as in the EconBiz, led HiAGM to run out of memory on a 40 GB RAM NVIDIA A100 HGX GPU. The presence of single-head attention layers in aMLP did not consistently improve performance compared to gMLP. While attention increased the sample-based F1 score by a few percent on the EconBiz and GoEmotions datasets, performance was the same or even less than that of gMLP on other datasets. Similarly, HGCLR and BERT+HiMatch that use BERT in conjunction with a hierarchy-processing graph-based model fail to outperform a simple pre-trained language model that does not make use of the class hierarchy.

Furthermore, an ablation study by Wang et al. [183] on their HGCLR method confirms our findings that using synthetically generated graphs is limited in improving text classification tasks. The authors have shown that removing the graph encoder does reduce the performance by about 1 point only (Micro/Macro-F1) [183]. We observe that using other methods, especially including transformer models in graph-based methods, improves the results much more. Similarly, Younes et al. [206] also provide empirical results that an explicit graph encoder is not needed for hierarchical text classification.

8.5 BERT Baselines are Often Undertuned

We found that BERT baselines are often undertuned in the literature. This can be declared as "baseline nerfing" , which may be accidental [81]. We hope that our comprehensive quantitative comparison sheds new light on the various proposed methods with solid BERT baselines. What we

argue to be particularly problematic is omitting BERT baselines as soon as some prior work has a marginal advantage over BERT, as this practice prohibits readers from properly contextualizing the results, e. g., when the new method is also only marginally better than BERT. Based on the results of our quantitative comparison, we argue that simple baselines such as BERT, an MLP, or an SVM should not be omitted in text classification.

8.6 Single-label vs. Multi-label Text Classification

We reflect on similarities and differences between single-label and multi-label text classification. Regarding the methods used for both tasks, i. e., single-label and multi-label classification, the best results are achieved by the fine-tuned transformer models. WideMLP gives comparable and sometimes better performance than many other recent models. Our results show that WideMLP can be considered a strong baseline for both single-label and multi-label classification tasks.

Another interesting observation can be made on the sentiment prediction dataset. In the single-label setup, BERT outperforms WideMLP on the MR dataset with the largest margin compared to other datasets. The same can be observed for the GoEmotions dataset in the multi-label case, where WideMLP achieves the worst performance across all models and the highest margin compared to BERT regarding all datasets. This shows that BoW-based MLP models might be at a disadvantage in sentiment prediction compared to sequence-based models. Note that most graph-based methods also discard word order when setting up the graph [45], except for models that combine the GNN with a sequence-based model, i. e., commonly a transformer.

8.7 Specific Aspects

In addition to the general discussion about the models' performance on text classification, we found several interesting aspects worth separate consideration.

Word Order. The main difference between bag-of-words and sequence-based models is whether models can capture word order information. BoW models discard word order entirely and yield good results. However, word order seems to be more important for sentiment-related tasks (such as the MR and GoEmotions datasets) than for topical classification tasks. In an extensive study, Conneau et al. [23] showed that memorizing word content (which words appear at all) is most indicative of performance on downstream tasks, among other linguistic properties. Sinha et al. [154] have experimented with pre-training BERT by disabling word order during pre-training and show that it makes surprisingly little difference for fine-tuning. In their study, word order is preserved during fine-tuning. Galke and Scherp [45] have experimented with the complementary setting of fine-tuning a standard BERT model without word order. The results show that deactivating position encoding and training on shuffled inputs does not increase the performance. Therefore, the strength of bag-of-words models can not solely be attributed to increased sample efficiency.

Our results confirm the notion that word order matters little for classifying documents into topics. Other NLP tasks such as question answering [141] or natural language inference [176] can also be regarded as instances of text classification. Here, the positional information is more important than it is in topic classification. In this case, we expect BoW-based models to perform worse than sequence-based models. This is also supported by our results on sentiment analysis, where the margin between bag-of-words-based models and pre-trained language models is the largest.

Although gMLP and aMLP models make use of positional information of the input, they fail to outperform the BoW-based MLP. The reason is that there are no pre-trained models available. This highlights the need for task-agnostic pre-training in sequence models and the cost-benefit of using

simpler models trained from scratch for text classification. Evaluating pre-trained gMLP and aMLP models remains future work.

Document Length. Notable on 20ng is also the performance of CogLTX, a variant of BERT specifically designed for long text [33]. CogLTX with a fine-tuned RoBERTa (for 4 epochs) reaches an accuracy of 87.0 on 20ng. This is only similar to the performance of a BERT-base with 87.21. This suggests that the extra features of CogLTX have no effect on the 20ng dataset. It may also be the case, citing CogLTX itself, that "for most NLP tasks, a few key sentences in the text hold sufficient and necessary information to fulfill the task" [33]. Subsequently, Fiok et al. [42] experiment with different truncation techniques for long text, sometimes leading to an advantage over first-512-tokens truncation. We leave studying the applicability of further long-range transformer models for text classification, e. g., [6, 42], as part of future work. Among our datasets, 20ng is the only one where many documents exceed the 512-token threshold.

Reinforcement Learning. Chai et al. [18] propose an approach using reinforcement learning for text classification, where the idea is to use large language models and learn descriptions of classes from data. The two best-performing variants are learning descriptions by extraction and abstraction. The results on the single-label dataset 20ng are good with an accuracy of 84.4 (extractive) and 84.6 (abstractive) methods but not competitive with the state of the art (both numbers not shown in Table 3 for brevity). Notably, Chai et al. [18] also report the lowest BERT-base score for 20ng with an accuracy of 83.1, which is more than one point less than our TF-IDF + WideMLP. Similarly, for the multi-label case on the R21578 dataset, the accuracy of the reinforcement learning method is good but not competitive to the state of the art.

8.8 Further discussions

Yuan et al. [214] report scores on the 20ng dataset for an SVM with 86 points and their *k*NN reaches 82. The authors claim that MSVM-*k*NN, a stacking of an SVM with subsequent *k*NN for documents where the SVM cannot make a decision, achieves a score of 90 for the 20ng dataset [214]. A similar observation was made with MHGAT, which obtained a score of 92.68 on the 20ng dataset. However, it is unclear what train-test split is used and if the metadata of the newsgroup posts, such as headers, footers, etc., were employed. The latter is an important parameter, as shown by the recent comparison of SVMs with pre-trained language models by Wahba et al. [175]. The authors report a performance boost of 17% when considering the metadata. Note, the results on the 20ng dataset reported by [175] are not comparable as an 80:20 split was used instead of the standard benchmark dataset split. Thus, we omit these specific numbers from the table but instead run several SVM variants ourselves on the datasets of our quantitative comparison.

9 LIMITATIONS

9.1 Dataset Selection

A limitation of our quantitative comparison is the selection of datasets on which we base this comparative survey. Although the selection of specific datasets could potentially bias the comparison, we chose the most common datasets for maximum coverage of the methods. To fill gaps in the literature, we run additional experiments with numerous methods on the selected datasets. Running these experiments is particularly important for multi-label classification, where the datasets are less standardized than in single-label classification. The current dataset collection was made to ensure a broad coverage of approaches for single-label text classification, multi-label text classification, and hierarchical text classification.

The choice of datasets in the multi-label text classification literature is more scattered than in the single-label case, which harms comparability. However, we have included the most prominent multi-label datasets, such as NYT or RCV1-V2, and also include datasets that go beyond the news domain, such as EconBiz, DBPedia, and even non-topical classification tasks, such as GoEmotions. For maximum comparability, we have reported three variants of the F measure in our own experiments.

We further emphasize that the experimental datasets are limited to English. While word order is important in the English language, it is notable that methods that discard word order still work very well for topical text classification. We assume that BoW-based models perform even better for languages with a richer morphology, where word order is less important [119]. It would be interesting to see to which extent our results of comparing BoW-based vs. sequence-based vs. graph-based vs. hierarchy-based methods for text classification transfer to other languages. Towards this direction, Gonzalez-Carvajal and Garrido-Merchan [48] show that BERT outperforms classic TF-IDF BoW approaches on English, Chinese, and Portuguese text classification datasets. But other methods are yet to be considered. Another direction is to consider specifically designed text classifiers for a single language, e. g., in such Chinese [50] where methods are tailored to the characteristics of Chinese characters, words, and radical information. Besides analyzing datasets in other languages, there is undoubtedly room for an even larger coverage of datasets in future work [9].

9.2 Pre-trained Attention-free Language Models

Regarding the sequential MLP-based models gMLP and aMLP, our study is limited to training them from scratch without large-scale pre-training. We expect these models to perform much better if they were pre-trained on large unlabeled text corpora in the same way as the transformer-based models. Unfortunately, such pre-trained gMLP/aMLP models were not publicly available. Pre-training and evaluating gMLP/aMLP models on large text corpora is a promising direction of future research, where it needs to be validated that they are on par with transformer-based models.

9.3 Data Contamination in Large Language Models

A big problem in the context of using generative language models is also that they are trained including on a lot of benchmark datasets [4]. It is not (always) clear if certain test sets are included in the language models pre-training data.

10 FUTURE DIRECTIONS AND CHALLENGES

Our comparison allow us to highlight some promising future directions for text classification.

10.1 Fine-tuning large language models

As argued in the previous section, the advantage of SLMs over generative LLMs mainly stems from the fact that SLMs are fine-tuned on the entire training set, while generative language models merely do in-context learning with few examples. However, large language models can also be fine-tuned, as exemplified by [92]: A Llama-2 model [169] can be successfully fine-tuned for text classification. This strengthens the view that the main distinction is whether the model is fine-tuned or not and hints at fine-tuning of generative language models as a promising direction of future work for text classification.

10.2 Scaling masked language models vs. unmasking causal language models during fine-tuning

A different option is to increase the model size of masked language models. That is, train larger versions of BERT on modern dataset collections. However, masked language models have lower

"throughput" than causal language models. This is because causal language models obtain a training signal from every token, whereas masked language models only receive a training signal from masked tokens. Still, encoding left and right context has shown to be important for text classification performance [92]. It remains unclear which strategy is better: training a large masked language model or unmasking a large causal language model during fine-tuning.

10.3 Large language models for multi-label text classification

Large language models for hierarchical and multi-label text classification raises interesting questions, how to feed the hierarchy into the input context of a language model, e. g., as in [39]. A main challenge in multi-label classification is that the context window of LLM's is oftentimes not large enough to fit even one example for each class, let alone examples for combinations of multiple classes. Nevertheless, compressing a multi-label training set into a well-performing few-shot example prompt is an interesting direction of future research.

10.4 Further Directions

Future work could also expand on hierarchy-based models. Techniques to learn independent thresholds for each class as proposed by Pellegrini and Masquelier [128] or Benedikt et al. [7] could further improve the results.

Another interesting yet challenging setting is few-shot classification as in prompt-based large language models [13]. It would be interesting to compare end-task-aware pre-training against fine-tuning after pre-training [30].

11 CONCLUSION

Returning to the question of whether we are making much progress in text classification, our extensive comparison has revealed a worrying state of affairs. Despite tremendous effort, none of the recently proposed methods that operate on graphs provides a benefit over fine-tuning a pre-trained language model, regardless of whether the graph is derived from the text or if a hierarchy is provided with a dataset. Even worse, many new approaches fail to outperform straightforward baselines, such as an SVM or a multilayer perception. Moreover, despite the astounding performance of LLMs in zero-shot and few-shot prompting, the best performance is achieved through fine-tuning, for which small language models seem to be sufficient.

We argue that future research in text classification should employ at least two baselines: a pre-trained transformer model and a wide multi-layer perceptron. The wide multilayer perceptron enhanced with today's best practices does not require much tuning and scores consistently high in topic classification tasks, being even the strongest model on the hardest multi-label dataset. Nevertheless, pre-trained transformers remain state of the art and are, besides the mentioned exception, only outperformed by approaches that use a pre-trained transformer as a component in their architecture.

Our study immediately impacts practitioners seeking to employ robust text classification models in research projects and industrial operational environments. Our recommendation to practitioners is to use a pre-trained language model when feasible, i. e., when sufficient computing power is available, and otherwise resort to a bag-of-words WideMLP as a well-tested solid model that further has an easier time processing long texts.

REFERENCES

[1] Victor Kwaku Agbesi, Wenyu Chen, Sophyani Banaamwini Yussif, Chiagoziem C. Ukwuoma, Yeong Hyeon Gu, and Mugahed A. Al-antari. 2024. MuTCELM: An optimal multi-TextCNN-based ensemble learning for text classification. *Heliyon* 10, 19 (2024), e38515. https://doi.org/10.1016/j.heliyon.2024.e38515

[2] Joseph Attieh and Joe Tekli. 2023. Supervised term-category feature weighting for improved text classification. *Knowledge-Based Systems* 261 (2023), 110215. https://doi.org/10.1016/j.knosys.2022.110215

[3] K. Bache and M. Lichman. 2013. UCI Machine Learning Repository. http://archive.ics.uci.edu/ml University of California, Irvine, School of Information and Computer Sciences.

[4] Simone Balloccu, Patrícia Schmidtová, Mateusz Lango, and Ondrej Dusek. 2024. Leak, Cheat, Repeat: Data Contamination and Evaluation Malpractices in Closed-Source LLMs. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, Yvette Graham and Matthew Purver (Eds.). Association for Computational Linguistics, 67–93.

[5] Markus Bayer, Marc-André Kaufhold, and Christian Reuter. 2023. A Survey on Data Augmentation for Text Classification. *ACM Comput. Surv.* 55, 7 (2023), 146:1–146:39.

[6] Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The Long-Document Transformer. *CoRR* abs/2004.05150 (2020). arXiv:2004.05150

[7] Gabriel Bénédict, Vincent Koops, Daan Odijk, and Maarten de Rijke. 2021. sigmoidF1: A Smooth F1 Score Surrogate Loss for Multilabel Classification. *CoRR* abs/2108.10566 (2021). arXiv:2108.10566

[8] Benjamin Bergner, Andrii Skliar, Amelie Royer, Tijmen Blankevoort, Yuki M. Asano, and Babak Ehteshami Bejnordi. 2024. Think Big, Generate Quick: LLM-to-SLM for Fast Autoregressive Decoding. *CoRR* abs/2402.16844 (2024). https://doi.org/10.48550/ARXIV.2402.16844 arXiv:2402.16844

[9] K. Bhatia, K. Dahiya, H. Jain, P. Kar, A. Mittal, Y. Prabhu, and M. Varma. 2016. The extreme classification repository: Multi-label datasets and code. http://manikvarma.org/downloads/XC/XMLRepository.html

[10] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet Allocation. *J. Mach. Learn. Res.* 3 (2003), 993–1022.

[11] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomás Mikolov. 2017. Enriching Word Vectors with Subword Information. *Trans. Assoc. Comput. Linguistics* 5 (2017), 135–146.

[12] Claudia Breazzano, Danilo Croce, and Roberto Basili. 2021. Multi-task and Generative Adversarial Learning for Robust and Sustainable Text Classification. In *AIxIA 2021 - Advances in Artificial Intelligence - 20th International Conference of the Italian Association for Artificial Intelligence, Virtual Event, December 1-3, 2021, Revised Selected Papers (Lecture Notes in Computer Science, Vol. 13196)*. Springer, 228–244. https://doi.org/10.1007/978-3-031-08421-8_16

[13] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems 33*.

[14] Martin Juan José Bucher and Marco Martini. 2024. Fine-Tuned 'Small' LLMs (Still) Significantly Outperform Zero-Shot Generative AI Models in Text Classification. *arXiv:2406.08660* (2024).

[15] Margarita Bugueño and Gerard de Melo. 2023. Connecting the Dots: What Graph-Based Text Representations Work Best for Text Classification Using Graph Neural Networks? *arXiv:2305.14578* (2023).

[16] Sérgio Canuto, Daniel Xavier Sousa, Marcos André Gonçalves, and Thierson Couto Rosa. 2018. A Thorough Evaluation of Distance-Based Meta-Features for Automated Text Classification. *IEEE Transactions on Knowledge and Data Engineering* 30, 12 (Dec. 2018), 2242–2256. https://doi.org/10.1109/TKDE.2018.2820051

[17] Youngjin Chae and Thomas Davidson. 2023. Large Language Models for Text Classification: From Zero-Shot Learning to Fine-Tuning. *SocArXiv* (Aug 2023). https://doi.org/10.31235/osf.io/sthwk SocArXiv Preprint, https://osf.io/preprints/socarxiv/sthwk.

[18] Duo Chai, Wei Wu, Qinghong Han, Fei Wu, and Jiwei Li. 2020. Description Based Text Classification with Reinforcement Learning. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event (Proceedings of Machine Learning Research, Vol. 119)*. PMLR, 1371–1382. http://proceedings.mlr.press/v119/chai20a.html

[19] Ilias Chalkidis, Manos Fergadiotis, Sotiris Kotitsas, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. An Empirical Study on Large-Scale Multi-Label Text Classification Including Few and Zero-Shot Labels. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Online, 7503–7515. https://doi.org/10.18653/v1/2020.emnlp-main.607

[20] Haibin Chen, Qianli Ma, Zhenxi Lin, and Jiangyue Yan. 2021. Hierarchy-aware Label Semantics Matching Network for Hierarchical Text Classification. In *Proceedings of the 59th Annual Meeting of the Association for Computational*

Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021. Association for Computational Linguistics, 4370–4379. https://doi.org/10.18653/v1/2021.acl-long.337

[21] Huiyao Chen, Yu Zhao, Zulong Chen, Mengjia Wang, Liangyue Li, Meishan Zhang, and Min Zhang. 2024. Retrieval-style In-Context Learning for Few-shot Hierarchical Text Classification. *CoRR* abs/2406.17534 (2024). https://doi.org/10.48550/ARXIV.2406.17534 arXiv:2406.17534

[22] Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020.* OpenReview.net. https://openreview.net/forum?id=r1xMH1BtvB

[23] Alexis Conneau, Germán Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single \$&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers.* Association for Computational Linguistics, 2126–2136. https://doi.org/10.18653/v1/P18-1198

[24] Alexis Conneau, Holger Schwenk, Loïc Barrault, and Yann LeCun. 2017. Very Deep Convolutional Networks for Text Classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 1: Long Papers.* Association for Computational Linguistics, 1107–1116. https://doi.org/10.18653/v1/e17-1104

[25] Thomas M Cover and Joy A Thomas. 1991. *Elements of Information Theory.* Wiley. 413 pages.

[26] Maurizio Ferrari Dacrema, Paolo Cremonesi, and Dietmar Jannach. 2019. Are we really making much progress? A worrying analysis of recent neural recommendation approaches. In *Proceedings of the 13th ACM Conference on Recommender Systems, RecSys 2019, Copenhagen, Denmark, September 16-20, 2019.* ACM, 101–109. https://doi.org/10.1145/3298689.3347058

[27] Le Dai, Yu Yin, Enhong Chen, and Hui Xiong. 2024. Unifying Graph Retrieval and Prompt Tuning for Graph-Grounded Text Classification. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2024, Washington DC, USA, July 14-18, 2024*, Grace Hui Yang, Hongning Wang, Sam Han, Claudia Hauff, Guido Zuccon, and Yi Zhang (Eds.). ACM, 2682–2686. https://doi.org/10.1145/3626772.3657934

[28] Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan S. Cowen, Gaurav Nemade, and Sujith Ravi. 2020. GoEmotions: A Dataset of Fine-Grained Emotions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020.* Association for Computational Linguistics, 4040–4054. https://doi.org/10.18653/v1/2020.acl-main.372

[29] Zhaoyang Deng, Chenxiang Sun, Guoqiang Zhong, and Yuxu Mao. 2022. Text Classification with Attention Gated Graph Neural Network. *Cognitive Computation* 14, 4 (July 2022), 1464–1473. https://doi.org/10.1007/s12559-022-10017-3

[30] Lucio M Dery, Paul Michel, Ameet Talwalkar, and Graham Neubig. 2021. Should we be pre-training? an argument for end-task aware training as an alternative. *CoRR* abs/2109.07437 (2021). arXiv:2109.07437

[31] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *North American Chapter of the ACL: Human Language Technologies, NAACL-HLT 2019.* ACL, 4171–4186. https://doi.org/10.18653/v1/n19-1423

[32] Kaize Ding, Jianling Wang, Jundong Li, Dingcheng Li, and Huan Liu. 2020. Be More with Less: Hypergraph Attention Networks for Inductive Text Classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020.* Association for Computational Linguistics, 4927–4936. https://doi.org/10.18653/v1/2020.emnlp-main.399

[33] Ming Ding, Chang Zhou, Hongxia Yang, and Jie Tang. 2020. CogLTX: Applying BERT to Long Texts. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual.* https://proceedings.neurips.cc/paper/2020/hash/96671501524948bc3937b4b30d0e57b9-Abstract.html

[34] Gregor Donabauer and Udo Kruschwitz. 2024. Token-Level Graphs for Short Text Classification. (2024). arXiv:2412.12754 [cs.IR] https://arxiv.org/abs/2412.12754

[35] Hang Dong, Víctor Suárez-Paniagua, William Whiteley, and Honghan Wu. 2021. Explainable automated coding of clinical notes using hierarchical label-wise attention networks and label embedding initialisation. *J. Biomed. Informatics* 116 (2021), 103728. https://doi.org/10.1016/j.jbi.2021.103728

[36] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021.* OpenReview.net. https://openreview.net/forum?id=YicbFdNTTy

[37] José Marcio Duarte and Lilian Berton. 2023. A review of semi-supervised learning for text classification. *Artificial Intelligence Review* (2023), 1–69.

[38] Aleksandra Edwards and Jose Camacho-Collados. 2024. Language Models for Text Classification: Is In-Context Learning Enough?. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, Nicoletta Calzolari, Min-Yen Kan, Veronique Hoste, Alessandro Lenci, Sakriani Sakti, and Nianwen Xue (Eds.). ELRA and ICCL, Torino, Italia, 10058–10072. https://aclanthology.org/2024.lrec-main.879

[39] Bahare Fatemi, Jonathan Halcrow, and Bryan Perozzi. 2024. Talk like a Graph: Encoding Graphs for Large Language Models. In *The Twelfth International Conference on Learning Representations*. https://openreview.net/forum?id=IuXR1CCrSi

[40] Matthias Fey, Jan Eric Lenssen, Frank Weichert, and Jure Leskovec. 2021. GNNAutoScale: Scalable and Expressive Graph Neural Networks via Historical Embeddings. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event (Proceedings of Machine Learning Research, Vol. 139)*. PMLR, 3294–3304. http://proceedings.mlr.press/v139/fey21a.html

[41] John Fields, Kevin Chovanec, and Praveen Madiraju. 2024. A Survey of Text Classification with Transformers: How Wide? How Large? How Long? How Accurate? How Expensive? How Safe? *IEEE Access* (2024), 1–1. https://doi.org/10.1109/ACCESS.2024.3349952

[42] Krzysztof Fiok, Waldemar Karwowski, Edgar Gutierrez-Franco, Mohammad Reza Davahli, Maciej Wilamowski, Tareq Z. Ahram, Awad M. Aljuaid, and Jozef M. Zurada. 2021. Text Guide: Improving the Quality of Long Text Classification by a Text Selection Method Based on Feature Importance. *IEEE Access* 9 (2021), 105439–105450. https://doi.org/10.1109/ACCESS.2021.3099758

[43] Lukas Galke, Andor Diera, Bao Xin Lin, Bhakti Khera, Tim Meuser, Tushar Singhal, Fabian Karl, and Ansgar Scherp. 2023. Are We Really Making Much Progress? Bag-of-Words vs. Sequence vs. Graph vs. Hierarchy for Single- and Multi-Label Text Classification. *CoRR* abs/2204.03954 (2023). arXiv:2204.03954

[44] Lukas Galke, Florian Mai, Alan Schelten, Dennis Brunsch, and Ansgar Scherp. 2017. Using Titles vs. Full-text as Source for Automated Semantic Document Annotation. In *Knowledge Capture Conference, K-CAP 2017*. ACM, 20:1–20:4. https://doi.org/10.1145/3148011.3148039

[45] Lukas Galke and Ansgar Scherp. 2022. Bag-of-Words vs. Graph vs. Sequence in Text Classification: Questioning the Necessity of Text-Graphs and the Surprising Strength of a Wide MLP. In *60th Annual Meeting of the Association for Computational Linguistics, ACL 2022*. Association for Computational Linguistics, 4038–4051. https://doi.org/10.18653/v1/2022.acl-long.279

[46] Lingyu Gao. 2024. Harnessing the Intrinsic Knowledge of Pretrained Language Models for Challenging Text Classification Settings. *CoRR* abs/2408.15650 (2024). https://doi.org/10.48550/ARXIV.2408.15650 arXiv:2408.15650

[47] Samujjwal Ghosh, Subhadeep Maji, and Maunendra Sankar Desarkar. 2021. Supervised Graph Contrastive Pretraining for Text Classification. *CoRR* abs/2112.11389 (2021). arXiv:2112.11389

[48] Santiago González-Carvajal and Eduardo C. Garrido-Merchán. 2020. Comparing BERT against traditional machine learning text classification. *CoRR* abs/2005.13012 (2020). arXiv:2005.13012

[49] Albert Gu and Tri Dao. 2023. Mamba: Linear-Time Sequence Modeling with Selective State Spaces. *arXiv:2312.00752* (2023).

[50] Jiabao Guo, Bo Zhao, Hui Liu, Yifan Liu, and Qian Zhong. 2023. Supervised Contrastive Learning with Term Weighting for Improving Chinese Text Classification. *Tsinghua Science and Technology* 28, 1 (2023), 59–68. https://doi.org/10.26599/TST.2021.9010079

[51] Suchin Gururangan, Tam Dang, Dallas Card, and Noah A. Smith. 2019. Variational Pretraining for Semi-supervised Text Classification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 5880–5894. https://doi.org/10.18653/v1/P19-1590

[52] William L. Hamilton. 2020. *Graph Representation Learning*. Morgan and Claypool.

[53] Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. DeBERTaV3: Improving DeBERTa using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net. https://openreview.net/forum?id=sE7-XhLxHA

[54] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. DeBERTa: Decoding-enhanced BERT with Disentangled Attention. *CoRR* abs/2006.03654 (2021). arXiv:2006.03654

[55] Bairu Hou, Joe O'Connor, Jacob Andreas, Shiyu Chang, and Yang Zhang. 2023. PromptBoosting: Black-Box Text Classification with Ten Forward Passes. In *Proceedings of the 40th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 202)*, Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (Eds.). PMLR, 13309–13324. https://proceedings.mlr.press/v202/hou23b.html

[56] Beizhe Hu, Qiang Sheng, Juan Cao, Yuhui Shi, Yang Li, Danding Wang, and Peng Qi. 2024. Bad Actor, Good Advisor: Exploring the Role of Large Language Models in Fake News Detection. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, Michael J. Wooldridge, Jennifer G. Dy, and Sriraam Natarajan (Eds.). AAAI Press, 22105–22113. https://doi.org/10.1609/AAAI.V38I20.30214

[57] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *The Tenth International Conference on Learning Representations, ICLR 2022*. OpenReview.net. https://openreview.net/forum?id=nZeVKeeFYf9

[58] Linmei Hu, Zeyi Liu, Ziwang Zhao, Lei Hou, Liqiang Nie, and Juanzi Li. 2024. A Survey of Knowledge Enhanced Pre-Trained Language Models. *IEEE Trans. Knowl. Data Eng.* 36, 4 (2024), 1413–1430. https://doi.org/10.1109/TKDE.2023.3310002

[59] Lianzhe Huang, Dehong Ma, Sujian Li, Xiaodong Zhang, and Houfeng Wang. 2019. Text Level Graph Neural Network for Text Classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*. Association for Computational Linguistics, 3442–3448. https://doi.org/10.18653/v1/D19-1345

[60] Yen-Hao Huang, Yi-Hsin Chen, and Yi-Shin Chen. 2022. ConTextING: Granting Document-Wise Contextual Embeddings to Graph Neural Networks for Inductive Text Classification. In *Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022*. International Committee on Computational Linguistics, 1163–1168. https://aclanthology.org/2022.coling-1.100

[61] Yi Huang, Buse Giledereli, Abdullatif Köksal, Arzucan Özgür, and Elif Ozkirimli. 2021. Balancing Methods for Multi-label Text Classification with Long-Tailed Class Distribution. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*. Association for Computational Linguistics, 8153–8161. https://doi.org/10.18653/v1/2021.emnlp-main.643

[62] Mohit Iyyer, Varun Manjunatha, Jordan L. Boyd-Graber, and Hal Daumé III. 2015. Deep Unordered Composition Rivals Syntactic Methods for Text Classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*. The Association for Computer Linguistics, 1681–1691. https://doi.org/10.3115/v1/p15-1162

[63] Saman Jamshidi, Mahin Mohammadi, Saeed Bagheri, Hamid Esmaeili Najafabadi, Alireza Rezvanian, Mehdi Gheisari, Mustafa Ghaderzadeh, Amir Shahab Shahabi, and Zongda Wu. 2024. Effective text classification using BERT, MTM LSTM, and DT. *Data Knowl. Eng.* 151, C (jul 2024), 17 pages. https://doi.org/10.1016/j.datak.2024.102306

[64] Saman Jamshidi, Mahin Mohammadi, Saeed Bagheri, Hamid Esmaeili Najafabadi, Alireza Rezvanian, Mehdi Gheisari, Mustafa Ghaderzadeh, Amir Shahab Shahabi, and Zongda Wu. 2024. Effective text classification using BERT, MTM LSTM, and DT. *Data and Knowledge Engineering* 151 (2024), 102306. https://doi.org/10.1016/j.datak.2024.102306

[65] Xiangen Jia, Min Jiang, Yihong Dong, Feng Zhu, Haocai Lin, Yu Xin, and Huahui Chen. 2023. Multimodal heterogeneous graph attention network. *Neural Comput. Appl.* 35, 4 (2023), 3357–3372. https://doi.org/10.1007/S00521-022-07862-6

[66] Ting Jiang, Deqing Wang, Leilei Sun, Zhongzhi Chen, Fuzhen Zhuang, and Qinghong Yang. 2022. Exploiting Global and Local Hierarchies for Hierarchical Text Classification. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 4030–4039. https://aclanthology.org/2022.emnlp-main.268

[67] Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. TinyBERT: Distilling BERT for Natural Language Understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020 (Findings of ACL, Vol. EMNLP 2020)*. Association for Computational Linguistics, 4163–4174. https://doi.org/10.18653/v1/2020.findings-emnlp.372

[68] Yilun Jin, Wei Yin, Haoseng Wang, and Fang He. 2024. Capturing word positions does help: A multi-element hypergraph gated attention network for document classification. *Expert Syst. Appl.* 251 (2024), 124002. https://doi.org/10.1016/J.ESWA.2024.124002

[69] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomás Mikolov. 2017. Bag of Tricks for Efficient Text Classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 2: Short Papers*. Association for Computational Linguistics, 427–431. https://doi.org/10.18653/v1/e17-2068

[70] Ammar Ismael Kadhim. 2019. Survey on supervised machine learning techniques for automatic text classification. *Artif. Intell. Rev.* 52, 1 (2019), 273–292.

[71] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A Convolutional Neural Network for Modelling Sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014,*

June 22-27, 2014, Baltimore, MD, USA, Volume 1: Long Papers. The Association for Computer Linguistics, 655–665. https://doi.org/10.3115/v1/p14-1062

[72] Fabian Karl and Ansgar Scherp. 2023. Transformers are Short-Text Classifiers. In *Machine Learning and Knowledge Extraction - 7th IFIP TC 5, TC 12, WG 8.4, WG 8.9, WG 12.9 International Cross-Domain Conference, CD-MAKE 2023, Benevento, Italy, August 29 - September 1, 2023, Proceedings (Lecture Notes in Computer Science, Vol. 14065)*, Andreas Holzinger, Peter Kieseberg, Federico Cabitza, Andrea Campagner, A Min Tjoa, and Edgar R. Weippl (Eds.). Springer, 103–122. https://doi.org/10.1007/978-3-031-40837-3_7

[73] Siddhant Kharbanda, Atmadeep Banerjee, Devaansh Gupta, Akash Palrecha, and Rohit Babbar. 2023. InceptionXML: A Lightweight Framework with Synchronized Negative Sampling for Short Text Extreme Classification. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '23)*. Association for Computing Machinery, New York, NY, USA, 760–769. https://doi.org/10.1145/3539618.3591699

[74] Siddhant Kharbanda, Atmadeep Banerjee, Erik Schultheis, and Rohit Babbar. 2022. CascadeXML: Rethinking Transformers for End-to-end Multi-resolution Training in Extreme Multi-label Classification. In *Advances in Neural Information Processing Systems*, Vol. 35. 2074–2087.

[75] Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*. ACL, 1746–1751. https://doi.org/10.3115/v1/d14-1181

[76] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net. https://openreview.net/forum?id=SJU4ayYgl

[77] Kamran Kowsari, Donald E. Brown, Mojtaba Heidarysafa, Kiana Jafari Meimandi, Matthew S. Gerber, and Laura E. Barnes. 2017. HDLTex: Hierarchical Deep Learning for Text Classification. In *16th IEEE International Conference on Machine Learning and Applications, ICMLA 2017, Cancun, Mexico, December 18-21, 2017*. IEEE, 364–371. https://doi.org/10.1109/ICMLA.2017.0-134

[78] Kamran Kowsari, Kiana Jafari Meimandi, Mojtaba Heidarysafa, Sanjana Mendu, Laura E. Barnes, and Donald E. Brown. 2019. Text Classification Algorithms: A Survey. *Inf.* 10, 4 (2019), 150.

[79] Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Recurrent Convolutional Neural Networks for Text Classification. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA*. AAAI Press, 2267–2273. http://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9745

[80] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. *CoRR* abs/1909.11942 (2020). arXiv:1909.11942

[81] Gavin Leech, Juan J. Vazquez, Misha Yagudin, Niclas Kupper, and Laurence Aitchison. 2024. Questionable practices in machine learning. *arXiv:2407.12220* (2024).

[82] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, and Christian Bizer. 2015. DBpedia - A large-scale, multilingual knowledge base extracted from Wikipedia. *Semantic Web* 6 (2015), 167–195.

[83] Pierre Lepagnol, Thomas Gerald, Sahar Ghannay, Christophe Servan, and Sophie Rosset. 2024. Small Language Models are Good Too: An Empirical Study of Zero-Shot Classification. *arXiv:2404.11122* (2024).

[84] Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The Power of Scale for Parameter-Efficient Prompt Tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (Eds.). Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 3045–3059. https://doi.org/10.18653/v1/2021.emnlp-main.243

[85] David D. Lewis, Yiming Yang, Tony G. Rose, and Fan Li. 2004. RCV1: A New Benchmark Collection for Text Categorization Research. *J. Mach. Learn. Res.* 5 (dec 2004), 361–397.

[86] Mengyu Li, Yonghao Liu, Fausto Giunchiglia, Xiaoyue Feng, and Renchu Guan. 2024. Simple-Sampling and Hard-Mixup with Prototypes to Rebalance Contrastive Learning for Text Classification. *CoRR* abs/2405.11524 (2024). https://doi.org/10.48550/ARXIV.2405.11524 arXiv:2405.11524

[87] Qian Li, Hao Peng, Jianxin Li, Congying Xia, Renyu Yang, Lichao Sun, Philip S. Yu, and Lifang He. 2022. A Survey on Text Classification: From Traditional to Deep Learning. *ACM Trans. Intell. Syst. Technol.* 13, 2 (2022), 31:1–31:41. https://doi.org/10.1145/3495162

[88] Shengnan Li, Xiaoming Wu, Xiangzhi Liu, Xuqiang Xue, and Yang Yu. 2023. Joint Training Graph Neural Network for the Bidding Project Title Short Text Classification. In *Web and Big Data - 7th International Joint Conference, APWeb-WAIM 2023, Wuhan, China, October 6-8, 2023, Proceedings, Part I (Lecture Notes in Computer Science, Vol. 14331)*, Xiangyu Song, Ruyi Feng, Yunliang Chen, Jianxin Li, and Geyong Min (Eds.). Springer, 252–267. https://doi.org/10.1007/978-981-97-2303-4_17

[89] Xianghua Li, Xinyu Wu, Zheng Luo, Zhanwei Du, Zhen Wang, and Chao Gao. 2022. Integration of global and local information for text classification. *Neural Computing and Applications* (Aug. 2022). https://doi.org/10.1007/s00521-

         022-07727-y
[90]   Xianzhi Li, Xiaodan Zhu, Zhiqiang Ma, Xiaomo Liu, and Sameena Shah. 2023. Are ChatGPT and GPT-4 General-Purpose Solvers for Financial Text Analytics? An Examination on Several Typical Tasks. *arXiv:2305.05862* (2023).
[91]   Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard S. Zemel. 2016. Gated Graph Sequence Neural Networks. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*. http://arxiv.org/abs/1511.05493
[92]   Zongxi Li, Xianming Li, Yuzhang Liu, Haoran Xie, Jing Li, Fu lee Wang, Qing Li, and Xiaoqin Zhong. 2023. Label Supervised LLaMA Finetuning. *arXiv:2310.01208* (2023).
[93]   Wenxin Liang, Tingyu Zhang, Han Liu, and Feng Zhang. 2024. SELP: A Semantically-Driven Approach for Separated and Accurate Class Prototypes in Few-Shot Text Classification. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.). Association for Computational Linguistics, 9732–9741. https://aclanthology.org/2024.findings-acl.579
[94]   Yuxiao Lin, Yuxian Meng, Xiaofei Sun, Qinghong Han, Kun Kuang, Jiwei Li, and Fei Wu. 2021. BertGCN: Transductive Text Classification by Combining GNN and BERT. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021 (Findings of ACL, Vol. ACL/IJCNLP 2021)*. Association for Computational Linguistics, 1456–1462. https://doi.org/10.18653/v1/2021.findings-acl.126
[95]   lingfei wu, yu chen, kai shen, xiaojie guo, hanning gao, shucheng li, jian pei, and bo long. 2023. graph neural networks for natural language processing: a survey. *foundations and trends® in machine learning* 16, 2 (2023), 119–328. https://doi.org/10.1561/2200000096
[96]   Chun Liu, Hongguang Zhang, Kainan Zhao, Xinghai Ju, and Lin Yang. 2024. LLMEmbed: Rethinking Lightweight LLM's Genuine Function in Text Classification. *arXiv:2406.03725* (2024).
[97]   Hanxiao Liu, Zihang Dai, David R. So, and Quoc V. Le. 2021. Pay Attention to MLPs. *CoRR* abs/2105.08050 (2021). arXiv:2105.08050
[98]   Hui Liu, Danqing Zhang, Bing Yin, and Xiaodan Zhu. 2021. Improving Pretrained Models for Zero-shot Multi-label Text Classification through Reinforced Label Hierarchy Reasoning. *CoRR* abs/2104.01666 (2021). arXiv:2104.01666
[99]   Han Liu, Siyang Zhao, Xiaotong Zhang, Feng Zhang, Wei Wang, Fenglong Ma, Hongyang Chen, Hong Yu, and Xianchao Zhang. 2024. Liberating Seen Classes: Boosting Few-Shot and Zero-Shot Text Classification via Anchor Generation and Classification Reframing. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, Michael J. Wooldridge, Jennifer G. Dy, and Sriraam Natarajan (Eds.). AAAI Press, 18644–18652. https://doi.org/10.1609/AAAI.V38I17.29827
[100] Tengfei Liu, Yongli Hu, Boyue Wang, Yanfeng Sun, Junbin Gao, and Baocai Yin. 2022. Hierarchical Graph Convolutional Networks for Structured Long Document Classification. *IEEE Transactions on Neural Networks and Learning Systems* (2022), 1–15. https://doi.org/10.1109/TNNLS.2022.3185295
[101] Weiwei Liu, Xiaobo Shen, Haobo Wang, and Ivor W. Tsang. 2020. The Emerging Trends of Multi-Label Learning. *CoRR* abs/2011.11197 (2020). arXiv:2011.11197
[102] Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2022. P-Tuning: Prompt Tuning Can Be Comparable to Fine-tuning Across Scales and Tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (Eds.). Association for Computational Linguistics, Dublin, Ireland, 61–68. https://doi.org/10.18653/v1/2022.acl-short.8
[103] Xien Liu, Xinxin You, Xiao Zhang, Ji Wu, and Ping Lv. 2020. Tensor Graph Convolutional Networks for Text Classification. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*. AAAI Press, 8409–8416. https://ojs.aaai.org/index.php/AAAI/article/view/6359
[104] Yonghao Liu, Renchu Guan, Fausto Giunchiglia, Yanchun Liang, and Xiaoyue Feng. 2021. Deep Attention Diffusion Graph Neural Networks for Text Classification. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*. Association for Computational Linguistics, 8142–8152. https://doi.org/10.18653/v1/2021.emnlp-main.642
[105] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR* abs/1907.11692 (2019). arXiv:1907.11692
[106] Ye Liu, Kai Zhang, Zhenya Huang, Kehang Wang, Yanghai Zhang, Qi Liu, and Enhong Chen. 2023. Enhancing Hierarchical Text Classification through Knowledge Graph Integration. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki (Eds.). Association for Computational Linguistics, 5797–5810. https://doi.org/10.18653/V1/2023.FINDINGS-ACL.358

[107] Khang Ly, Yury Kashnitsky, Savvas Chamezopoulos, and Valeria V. Krzhizhanovskaya. 2024. Article Classification with Graph Neural Networks and Multigraphs. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation, LREC/COLING 2024, 20-25 May, 2024, Torino, Italy*, Nicoletta Calzolari, Min-Yen Kan, Véronique Hoste, Alessandro Lenci, Sakriani Sakti, and Nianwen Xue (Eds.). ELRA and ICCL, 1539–1547. https://aclanthology.org/2024.lrec-main.136

[108] Shengfei Lyu and Jiaqi Liu. 2020. Combine Convolution with Recurrent Networks for Text Classification. *CoRR* abs/2006.15795 (2020). arXiv:2006.15795

[109] Florian Mai, Lukas Galke, and Ansgar Scherp. 2018. Using Deep Learning for Title-Based Semantic Subject Indexing to Reach Competitive Performance to Full-Text. In *Proceedings of the 18th ACM/IEEE on Joint Conference on Digital Libraries, JCDL 2018, Fort Worth, TX, USA, June 03-07, 2018*. ACM, 169–178. https://doi.org/10.1145/3197026.3197039

[110] Yanxu Mao, Peipei Liu, Tiehan Cui, Congying Liu, and Datao You. 2024. Low-Resource Fast Text Classification Based on Intra-Class and Inter-Class Distance Calculation. (2024). arXiv:2412.09922 [cs.CL] https://arxiv.org/abs/2412.09922

[111] Julian McAuley and Jure Leskovec. 2013. Hidden Factors and Hidden Topics: Understanding Rating Dimensions with Review Text. In *Proceedings of the 7th ACM Conference on Recommender Systems* (Hong Kong, China) *(RecSys '13)*. Association for Computing Machinery, New York, NY, USA, 165–172. https://doi.org/10.1145/2507157.2507163

[112] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*. 3111–3119.

[113] Shervin Minaee, Nal Kalchbrenner, Erik Cambria, Narjes Nikzad, Meysam Chenaghlu, and Jianfeng Gao. 2022. Deep Learning-based Text Classification: A Comprehensive Review. *ACM Comput. Surv.* 54, 3 (2022), 62:1–62:40. https://doi.org/10.1145/3439726

[114] Alejandro Moreo, Andrea Esuli, and Fabrizio Sebastiani. 2020. Learning to Weight for Text Classification. *IEEE Trans. Knowl. Data Eng.* 32, 2 (2020), 302–316. https://doi.org/10.1109/TKDE.2018.2883446

[115] Alejandro Moreo, Andrea Esuli, and Fabrizio Sebastiani. 2020. Learning to Weight for Text Classification. *IEEE Transactions on Knowledge and Data Engineering* 32, 2 (2020), 302–316. https://doi.org/10.1109/TKDE.2018.2883446

[116] Ramesh Nallapati. 2004. Discriminative models for information retrieval. In *SIGIR 2004: Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Sheffield, UK, July 25-29, 2004*, Mark Sanderson, Kalervo Järvelin, James Allan, and Peter Bruza (Eds.). ACM, 64–71. https://doi.org/10.1145/1008992.1009006

[117] Jinseok Nam, Eneldo Loza Mencía, Hyunwoo J Kim, and Johannes Fürnkranz. 2017. Maximizing Subset Accuracy with Recurrent Neural Networks in Multi-label Classification. In *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.). Curran Associates, Inc., 5413–5423.

[118] Nobal Niraula, Samet Ayhan, Balaguruna Chidambaram, and Daniel Whyatt. 2024. Multi-Label Classification with Generative Large Language Models. In *2024 AIAA DATC/IEEE 43rd Digital Avionics Systems Conference (DASC)*. 1–7. https://doi.org/10.1109/DASC62030.2024.10748883

[119] Iga Nowak and Giosuè Baggio. 2016. The emergence of word order and morphology in compositional languages via multigenerational signaling games. *Journal of Language Evolution* 1, 2 (2016), 137–150. https://doi.org/10.1093/jole/lzw007

[120] Yukio Ohsawa, Nels E. Benson, and Masahiko Yachida. 1998. KeyGraph: Automatic Indexing by Co-Occurrence Graph Based on Building Construction Metaphor. In *Proceedings of the IEEE Forum on Research and Technology Advances in Digital Libraries*. IEEE Computer Society, 12–18.

[121] OpenAI. 2023. GPT-4 Technical Report. *CoRR* abs/2303.08774 (2023). https://doi.org/10.48550/ARXIV.2303.08774 arXiv:2303.08774

[122] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training Language Models to Follow Instructions with Human Feedback. *arXiv:2203.02155* (2022).

[123] Ankit Pal, Muru Selvakumar, and Malaikannan Sankarasubbu. 2020. MAGNET: Multi-Label Text Classification using Attention-based Graph Neural Network. In *Proceedings of the 12th International Conference on Agents and Artificial Intelligence, ICAART 2020, Volume 2, Valletta, Malta, February 22-24, 2020*. SCITEPRESS, 494–505. https://doi.org/10.5220/0008940304940505

[124] Mahak Pandia and Sahana Ramnath. 2022. Reproducing Bag-of-Words vs. Graph vs. Sequence in Text Classification: Questioning the Necessity of Text-Graphs and the Surprising Strength of a Wide MLP (ACL 2022). https://raw.githubusercontent.com/SahanaRamnath/bow-vs-graph-vs-seq-textclassification/main/Report_final_Pandia_Ramnath.pdf

[125]   Bo Pang and Lillian Lee. 2005. Seeing Stars: Exploiting Class Relationships for Sentiment Categorization with Respect to Rating Scales. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*. ACL, Ann Arbor, Michigan, 115–124. https://doi.org/10.3115/1219840.1219855

[126]   Bekir Parlak. 2023. A Novel Feature and Class-Based Globalization Technique for Text Classification. *Multimedia Tools and Applications* (April 2023). https://doi.org/10.1007/s11042-023-15459-x

[127]   Ajay Patel, Bryan Li, Mohammad Sadegh Rasooli, Noah Constant, Colin Raffel, and Chris Callison-Burch. 2023. Bidirectional Language Models Are Also Few-shot Learners. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net. https://openreview.net/pdf?id=wCFB37bzud4

[128]   Thomas Pellegrini and Timothée Masquelier. 2021. Fast threshold optimization for multi-label audio tagging using Surrogate gradient learning. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 651–655.

[129]   Bo Peng, Eric Alcaide, Quentin Anthony, Alon Albalak, Samuel Arcadinho, Stella Biderman, Huanqi Cao, Xin Cheng, Michael Chung, Leon Derczynski, Xingjian Du, Matteo Grella, Kranthi Gv, Xuzheng He, Haowen Hou, Przemyslaw Kazienko, Jan Kocon, Jiaming Kong, Bart\lomiej Koptyra, Hayden Lau, Jiaju Lin, Krishna Sri Ipsit Mantri, Ferdinand Mom, Atsushi Saito, Guangyu Song, Xiangru Tang, Johan Wind, Stanis\law Woźniak, Zhenyuan Zhang, Qinghua Zhou, Jian Zhu, and Rui-Jie Zhu. 2023. RWKV: Reinventing RNNs for the Transformer Era. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 14048–14077. https://doi.org/10.18653/v1/2023.findings-emnlp.936

[130]   Hao Peng, Jianxin Li, Senzhang Wang, Lihong Wang, Qiran Gong, Renyu Yang, Bo Li, Philip S. Yu, and Lifang He. 2021. Hierarchical Taxonomy-Aware and Attentional Graph Capsule RCNNs for Large-Scale Multi-Label Text Classification. *IEEE Transactions on Knowledge and Data Engineering* 33, 6 (2021), 2505–2519. https://doi.org/10.1109/TKDE.2019.2959991

[131]   Qiaojuan Peng, Xiong Luo, Yuqi Yuan, Fengbo Gu, Hailun Shen, and Ziyang Huang. 2025. A text classification method combining in-domain pre-training and prompt learning for the steel e-commerce industry. *International Journal of Web Information Systems* 21, 1 (2025), 96–119. https://doi.org/10.1108/IJWIS-09-2024-0277

[132]   Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*. ACL, 1532–1543. https://doi.org/10.3115/v1/d14-1162

[133]   Christos Petridis. 2024. Text Classification: Neural Networks VS Machine Learning Models VS Pre-trained Models. arXiv:2412.21022 [cs.LG] https://arxiv.org/abs/2412.21022

[134]   Yinhua Piao, Sangseon Lee, Dohoon Lee, and Sun Kim. 2022. Sparse Structure Learning via Graph Neural Networks for Inductive Document Classification. *Proceedings of the AAAI Conference on Artificial Intelligence* 36, 10 (June 2022), 11165–11173. https://doi.org/10.1609/aaai.v36i10.21366

[135]   Mohammadreza Qaraei, Sujay Khandagale, and Rohit Babbar. 2020. Why state-of-the-art deep learning barely works as good as a linear classifier in extreme multi-label text classification. In *28th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, ESANN 2020, Bruges, Belgium, October 2-4, 2020*. 223–228. https://www.esann.org/sites/default/files/proceedings/2020/ES2020-207.pdf

[136]   Muhammad Reza Qorib, Geonsik Moon, and Hwee Tou Ng. 2024. Are Decoder-Only Language Models Better than Encoder-Only Language Models in Understanding Word Meaning?. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.). Association for Computational Linguistics, 16339–16347. https://doi.org/10.18653/V1/2024.FINDINGS-ACL.967

[137]   ]radfordLanguageModelsAre Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. [n. d.]. Language Models Are Unsupervised Multitask Learners. ([n. d.]), 24.

[138]   Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *J. Mach. Learn. Res.* 21 (2020), 140:1–140:67.

[139]   Rahul Ragesh, Sundararajan Sellamanickam, Arun Iyer, Ramakrishna Bairi, and Vijay Lingam. 2021. HeteGCN: Heterogeneous Graph Convolutional Networks for Text Classification. In *WSDM '21, The Fourteenth ACM International Conference on Web Search and Data Mining, Virtual Event, Israel, March 8-12, 2021*. ACM, 860–868. https://doi.org/10.1145/3437963.3441746

[140]   Diardano Raihan. 2021. Deep Learning Techniques for Text Classification. https://towardsdatascience.com/deep-learning-techniques-for-text-classification-78d9dc40bf7c

[141]   Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100, 000+ Questions for Machine Comprehension of Text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*. The Association for Computational Linguistics, 2383–2392.

https://doi.org/10.18653/v1/d16-1264

[142] Manon Reusens, Alexander Stevens, Jonathan Tonglet, Johannes De Smedt, Wouter Verbeke, Seppe vanden Broucke, and Bart Baesens. 2024. Evaluating text classification: A benchmark study. *Expert Syst. Appl.* 254 (2024), 124302. https://doi.org/10.1016/J.ESWA.2024.124302

[143] Stuart Rose, Dave Engel, Nick Cramer, and Wendy Cowley. 2010. Automatic Keyword Extraction from Individual Documents. In *Text Mining. Applications and Theory*. John Wiley and Sons, Ltd, 1–20.

[144] Evan Sandhaus. 2008. The New York Times Annotated Corpus. Linguistic Data Consortium, Philadelphia, 6(12):e26752..

[145] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR* abs/1910.01108 (2019). arXiv:1910.01108

[146] R. Sarasu, K. K. Thyagharajan, and N. R. Shanker. 2023. SF-CNN: Deep Text Classification and Retrieval for Text Documents. *Intelligent Automation & Soft Computing* 35, 2 (2023), 1799–1813. https://doi.org/10.32604/iasc.2023.027429

[147] Fabrizio Sebastiani. 2002. Machine learning in automated text categorization. *ACM Comput. Surv.* 34, 1 (2002), 1–47. https://doi.org/10.1145/505282.505283

[148] Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. 2018. Pitfalls of Graph Neural Network Evaluation. *CoRR* abs/1811.05868 (2018). arXiv:1811.05868

[149] Faisal Shehzad and Dietmar Jannach. 2023. Everyone's a Winner! On Hyperparameter Tuning of Recommendation Models. In *Proceedings of the 17th ACM Conference on Recommender Systems, RecSys 2023, Singapore, Singapore, September 18-22, 2023*, Jie Zhang, Li Chen, Shlomo Berkovsky, Min Zhang, Tommaso Di Noia, Justin Basilico, Luiz Pizzato, and Yang Song (Eds.). ACM, 652–657. https://doi.org/10.1145/3604915.3609488

[150] Dinghan Shen, Guoyin Wang, Wenlin Wang, Martin Renqiang Min, Qinliang Su, Yizhe Zhang, Chunyuan Li, Ricardo Henao, and Lawrence Carin. 2018. Baseline Needs More Love: On Simple Word-Embedding-Based Models and Associated Pooling Mechanisms. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*. Association for Computational Linguistics, 440–450. https://doi.org/10.18653/v1/P18-1041

[151] Jiaming Shen, Wenda Qiu, Yu Meng, Jingbo Shang, Xiang Ren, and Jiawei Han. 2021. TaxoClass: Hierarchical Multi-Label Text Classification Using Only Class Names. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Online, 4239–4249. https://doi.org/10.18653/v1/2021.naacl-main.335

[152] Jinze Shi, Xiaoming Wu, Xiangzhi Liu, Wenpeng Lu, and Shu Li. 2022. Inductive Light Graph Convolution Network for Text Classification Based on Word-Label Graph. In *Intelligent Information Processing XI (IFIP Advances in Information and Communication Technology)*. Springer International Publishing, Cham, 42–55. https://doi.org/10.1007/978-3-031-03948-5_4

[153] Anjalee De Silva, Janaka L. Wijekoon, Rashini K. Liyanarachchi, Rrubaa Panchendrarajan, and Weranga Rajapaksha. 2024. AI Insights: A Case Study on Utilizing ChatGPT Intelligence for Research Paper Analysis. *CoRR* abs/2403.03293 (2024). https://doi.org/10.48550/ARXIV.2403.03293 arXiv:2403.03293

[154] Koustuv Sinha, Robin Jia, Dieuwke Hupkes, Joelle Pineau, Adina Williams, and Douwe Kiela. 2021. Masked Language Modeling and the Distributional Hypothesis: Order Word Matters Pre-training for Little. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*. Association for Computational Linguistics, 2888–2913. https://doi.org/10.18653/v1/2021.emnlp-main.230

[155] Comparing Specialised Small and General Large Language Models on Text Classification: 100 Labelled Samples to Achieve Break-Even Performance. 2024. Branislav Pecher and Ivan Srba and Maria Bielikova. *CoRR* abs/2402.12819 (2024). arXiv:2402.12819 https://doi.org/10.48550/arXiv.2402.12819

[156] Nikolaos Stylianou, Despoina Chatzakou, Theodora Tsikrika, Stefanos Vrochidis, and Ioannis Kompatsiaris. 2023. Domain-Aligned Data Augmentation for Low-Resource and Imbalanced Text Classification. In *Advances in Information Retrieval - 45th European Conference on Information Retrieval, ECIR 2023, Dublin, Ireland, April 2-6, 2023, Proceedings, Part II (Lecture Notes in Computer Science, Vol. 13981)*. Springer, 172–187. https://doi.org/10.1007/978-3-031-28238-6_12

[157] Guoying Sun, Jie Li, Yanan Cheng, and Zhaoxin Zhang. 2024. LMTCSG: Multilabel Text Classification Combining Sequence-Based and GNN-Based Features. *IEEE Transactions on Industrial Informatics* (2024), 1–9. https://doi.org/10.1109/TII.2024.3465596

[158] Xiaofei Sun, Xiaoya Li, Jiwei Li, Fei Wu, Shangwei Guo, Tianwei Zhang, and Guoyin Wang. 2023. Text Classification via Large Language Models. *arXiv:230508377* (2023).

[159] Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. 2019. ERNIE 2.0: A Continual Pre-training Framework for Language Understanding. *CoRR* abs/1907.12412 (2019). arXiv:1907.12412

[160] Zhongtian Sun, Anoushka Harit, Alexandra I. Cristea, Jialin Yu, Lei Shi, and Noura Al Moubayed. 2022. Contrastive Learning with Heterogeneous Graph Attention Networks on Short Text Classification. In *International Joint Conference on Neural Networks, IJCNN 2022, Padua, Italy, July 18-23, 2022*. IEEE, 1–6. https://doi.org/10.1109/IJCNN55064.2022.9892257

[161] Zhiqing Sun, Hongkun Yu, Xiaodan Song, Renjie Liu, Yiming Yang, and Denny Zhou. 2020. MobileBERT: A Compact Task-Agnostic BERT for Resource-Limited Devices. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*. Association for Computational Linguistics, 2158–2170. https://doi.org/10.18653/v1/2020.acl-main.195

[162] Changgeng Tan, Yuan Ren, and Chen Wang. 2023. An adaptive convolution with label embedding for text classification. *Appl. Intell.* 53, 1 (2023), 804–812. https://doi.org/10.1007/s10489-021-02702-x

[163] Zhipeng Tan, Jing Chen, Qi Kang, Mengchu Zhou, Abdullah Abusorrah, and Khaled Sedraoui. 2022. Dynamic Embedding Projection-Gated Convolutional Neural Networks for Text Classification. *IEEE Transactions on Neural Networks and Learning Systems* 33, 3 (2022), 973–982. https://doi.org/10.1109/TNNLS.2020.3036192

[164] Jian Tang, Meng Qu, and Qiaozhu Mei. 2015. PTE: Predictive Text Embedding through Large-scale Heterogeneous Text Networks. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, NSW, Australia, August 10-13, 2015*. ACM, 1165–1174. https://doi.org/10.1145/2783258.2783307

[165] Adane Nega Tarekegn, Mario Giacobini, and Krzysztof Michalak. 2021. A review of methods for imbalanced multi-label classification. *Pattern Recognit.* 118 (2021), 107965. https://doi.org/10.1016/j.patcog.2021.107965

[166] Thanakorn Thaminkaew, Piyawat Lertvittayakumjorn, and Peerapon Vateekul. 2024. Prompt-Based Label-Aware Framework for Few-Shot Multi-Label Text Classification. *IEEE Access* 12 (2024), 28310–28322. https://doi.org/10.1109/ACCESS.2024.3367994

[167] Thanakorn Thaminkaew, Piyawat Lertvittayakumjorn, and Peerapon Vateekul. 2024. Prompt-Based Label-Aware Framework for Few-Shot Multi-Label Text Classification. *IEEE Access* 12 (2024), 28310–28322. https://doi.org/10.1109/ACCESS.2024.3367994

[168] Ilya O. Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, Mario Lucic, and Alexey Dosovitskiy. 2021. MLP-Mixer: An all-MLP Architecture for Vision. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*. 24261–24272. https://proceedings.neurips.cc/paper/2021/hash/cba0a4ee5ccd02fda0fe3f9a3e7b89fe-Abstract.html

[169] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. arXiv:2307.09288 (July 2023). https://doi.org/10.48550/arXiv.2307.09288 arXiv:2307.09288 [cs]

[170] Quynh Tran, Krystsina Shpileuskaya, Elaine Zaunseder, Larissa Putzar, and Sven Blankenburg. 2022. Comparing the Robustness of Classical and Deep Learning Techniques for Text Classification. In *International Joint Conference on Neural Networks, IJCNN 2022, Padua, Italy, July 18-23, 2022*. IEEE, 1–10. https://doi.org/10.1109/IJCNN55064.2022.9892242

[171] Grigorios Tsoumakas and Ioannis Katakis. 2009. Multi-Label Classification: An Overview. *International Journal of Data Warehousing and Mining* 3 (09 2009), 1–13. https://doi.org/10.4018/jdwm.2007070101

[172] Niels van der Heijden, Ekaterina Shutova, and Helen Yannakoudakis. 2023. FewShotTextGCN: K-hop neighborhood regularization for few-shot learning on graphs. *CoRR* abs/2301.10481 (2023). arXiv:2301.10481

[173] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*. 5998–6008. https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html

[174] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net. https://openreview.net/forum?id=rJXMpikCZ

[175] Yasmen Wahba, Nazim H. Madhavji, and John Steinbacher. 2022. A Comparison of SVM against Pre-trained Language Models (PLMs) for Text Classification Tasks. *CoRR* abs/2211.02563 (2022). https://doi.org/10.48550/arXiv.2211.02563

arXiv:2211.02563

[176] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net. https://openreview.net/forum?id=rJ4km2R5t7

[177] Kunze Wang, Yihao Ding, and Soyeon Caren Han. 2023. Graph Neural Networks for Text Classification: A Survey. *CoRR* abs/2304.11534 (2023). arXiv:2304.11534

[178] Kunze Wang, Soyeon Caren Han, and Josiah Poon. 2022. InducT-GCN: Inductive Graph Convolutional Networks for Text Classification. In *26th International Conference on Pattern Recognition, ICPR 2022, Montreal, QC, Canada, August 21-25, 2022*. IEEE, 1243–1249. https://doi.org/10.1109/ICPR56361.2022.9956075

[179] Ruishuang Wang, Zhao Li, Jian Cao, Tong Chen, and Lei Wang. 2019. Convolutional Recurrent Neural Networks for Text Classification. In *International Joint Conference on Neural Networks, IJCNN 2019 Budapest, Hungary, July 14-19, 2019*. IEEE, 1–6. https://doi.org/10.1109/IJCNN.2019.8852406

[180] Xin Wang, Chao Wang, Haiyang Yang, Xingpeng Zhang, Qi Shen, Kan Ji, Yuhong Wu, and Huayi Zhan. 2022. KGAT: An Enhanced Graph-Based Model for Text Classification. In *Natural Language Processing and Chinese Computing - 11th CCF International Conference, NLPCC 2022, Guilin, China, September 24-25, 2022, Proceedings, Part I (Lecture Notes in Computer Science, Vol. 13551)*. Springer, 656–668. https://doi.org/10.1007/978-3-031-17120-8_51

[181] Yequan Wang, Aixin Sun, Jialong Han, Ying Liu, and Xiaoyan Zhu. 2018. Sentiment Analysis by Capsules. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, April 23-27, 2018*. ACM, 1165–1174. https://doi.org/10.1145/3178876.3186015

[182] Yaqing Wang, Song Wang, Quanming Yao, and Dejing Dou. 2021. Hierarchical Heterogeneous Graph Representation Learning for Short Text Classification. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*. Association for Computational Linguistics, 3091–3101. https://doi.org/10.18653/v1/2021.emnlp-main.247

[183] Zihan Wang, Peiyi Wang, Lianzhe Huang, Xin Sun, and Houfeng Wang. 2022. Incorporating Hierarchy into Text Encoder: a Contrastive Learning Approach for Hierarchical Text Classification. In *60th Annual Meeting of the Association for Computational Linguistics, ACL 2022*. ACL, 7109–7119. https://doi.org/10.18653/v1/2022.acl-long.491

[184] Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Nathan Cooper, Griffin Adams, Jeremy Howard, and Iacopo Poli. 2024. Smarter, Better, Faster, Longer: A Modern Bidirectional Encoder for Fast, Memory Efficient, and Long Context Finetuning and Inference. (2024). arXiv:2412.13663 [cs.CL] https://arxiv.org/abs/2412.13663

[185] Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. 2022. Finetuned Language Models are Zero-Shot Learners. In *International Conference on Learning Representations*. https://openreview.net/forum?id=gEZrGCozdqR

[186] Jia Wei and Xiangguo Sun. 2024. Study on text classification model combining BERT and convolutional neural network. *Mathematical Modeling and Algorithm Application* 2, 3 (Sep. 2024), 10–12. https://doi.org/10.54097/7h5bs772

[187] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. *arXiv:2201.11903* (2023).

[188] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (Eds.), Vol. 35. Curran Associates, Inc., 24824–24837. https://proceedings.neurips.cc/paper_files/paper/2022/file/9d5609613524ecf4f15af0f7b31abca4-Paper-Conference.pdf

[189] Felix Wu, Amauri H. Souza Jr., Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Q. Weinberger. 2019. Simplifying Graph Convolutional Networks. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA (Proceedings of Machine Learning Research, Vol. 97)*. PMLR, 6861–6871. http://proceedings.mlr.press/v97/wu19e.html

[190] Hui Wu, Yuanben Zhang, Zhonghe Han, Yingyan Hou, Lei Wang, Siye Liu, Qihang Gong, and Yunping Ge. 2024. Quartet Logic: A Four-Step Reasoning (QLFR) framework for advancing Short Text Classification. *CoRR* abs/2401.03158 (2024). https://doi.org/10.48550/ARXIV.2401.03158 arXiv:2401.03158

[191] Huiru Xiao, Xin Liu, and Yangqiu Song. 2019. Efficient Path Prediction for Semi-Supervised and Weakly Supervised Hierarchical Text Classification. *CoRR* abs/1902.09347 (2019). arXiv:1902.09347

[192] Zheng Xie, Yiqin Lv, Yiping Song, and Qi Wang. 2024. Data labeling through the centralities of co-reference networks improves the classification accuracy of scientific papers. *Journal of Informetrics* (2024). https://api.semanticscholar.org/CorpusID:267116237

[193] Xiantao Xu, Minghao Hu, Yongjie Wang, Wei Luo, Shilong Liu, ZhunChen Luo, and Yushan Tan. 2024. PLIClass: Weakly Supervised Text Classification with Iterative Training and Denoisy Inference. In *Artificial Neural Networks*

and Machine Learning – ICANN 2024, Michael Wand, Kristína Malinovská, Jürgen Schmidhuber, and Igor V. Tetko (Eds.). Springer Nature Switzerland, Cham, 292–305.

[194] Xuran Xu, Tong Zhang, Chunyan Xu, and Zhen Cui. 2021. Circulant Tensor Graph Convolutional Network for Text Classification. In *Pattern Recognition - 6th Asian Conference, ACPR 2021, Jeju Island, South Korea, November 9-12, 2021, Revised Selected Papers, Part I (Lecture Notes in Computer Science, Vol. 13188)*. Springer, 32–46. https://doi.org/10.1007/978-3-031-02375-0_3

[195] Bingxin Xue, Cui Zhu, Xuan Wang, and Wenjun Zhu. 2022. The Study on the Text Classification Based on Graph Convolutional Network and BiLSTM. In *ICCAI '22: 8th International Conference on Computing and Artificial Intelligence, Tianjin, China, March 18 - 21, 2022*. ACM, 323–331. https://doi.org/10.1145/3532213.3532261

[196] Heng Yang, Nan Wang, Lina Yang, Wei Liu, and Sili Wang. 2023. Research on the Automatic Subject-Indexing Method of Academic Papers Based on Climate Change Domain Ontology. *Sustainability* 15, 5 (Feb 2023), 3919. https://doi.org/10.3390/su15053919

[197] Jingfeng Yang, Hongye Jin, Ruixiang Tang, Xiaotian Han, Qizhang Feng, Haoming Jiang, Bing Yin, and Xia Hu. 2023. Harnessing the Power of LLMs in Practice: A Survey on ChatGPT and Beyond. (2023). arXiv:2304.13712 [cs.CL]

[198] Pengcheng Yang, Xu Sun, Wei Li, Shuming Ma, Wei Wu, and Houfeng Wang. 2018. SGM: Sequence Generation Model for Multi-label Classification. In *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, Emily M. Bender, Leon Derczynski, and Pierre Isabelle (Eds.). Association for Computational Linguistics, 3915–3926. https://aclanthology.org/C18-1330/

[199] Tianchi Yang, Linmei Hu, Chuan Shi, Houye Ji, Xiaoli Li, and Liqiang Nie. 2021. HGAT: Heterogeneous Graph Attention Networks for Semi-supervised Short Text Classification. *ACM Trans. Inf. Syst.* 39, 3 (2021), 32:1–32:29. https://doi.org/10.1145/3450352

[200] Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Graph Convolutional Networks for Text Classification. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019*. AAAI Press, 7370–7377. https://doi.org/10.1609/aaai.v33i01.33017370

[201] Zhihao Ye, Gongyao Jiang, Ye Liu, Zhiyong Li, and Jin Yuan. 2020. Document and Word Representations Generated by Graph Convolutional Network and BERT for Short Text Classification. In *ECAI 2020 - 24th European Conference on Artificial Intelligence, 29 August-8 September 2020, Santiago de Compostela, Spain, August 29 - September 8, 2020 - Including 10th Conference on Prestigious Applications of Artificial Intelligence (PAIS 2020) (Frontiers in Artificial Intelligence and Applications, Vol. 325)*. IOS Press, 2275–2281. https://doi.org/10.3233/FAIA200355

[202] Asaf Yehudai and Elron Bendel. 2024. When LLMs are Unfit Use FastFit: Fast and Effective Text Classification with Many Classes. *arXiv:2404.12365* (2024).

[203] Shu Yin, Peican Zhu, Xinyu Wu, Jiajin Huang, Xianghua Li, Zhen Wang, and Chao Gao. 2023. Integrating Information by Kullback–Leibler Constraint for Text Classification. *Neural Computing and Applications* (May 2023). https://doi.org/10.1007/s00521-023-08602-0

[204] Wenpeng Yin, Jamaal Hay, and Dan Roth. 2019. Benchmarking Zero-shot Text Classification: Datasets, Evaluation and Entailment Approach. *CoRR* abs/1909.00161 (2019). arXiv:1909.00161

[205] Ronghui You, Zihan Zhang, Ziye Wang, Suyang Dai, Hiroshi Mamitsuka, and Shanfeng Zhu. 2019. AttentionXML: Label Tree-based Attention-Aware Deep Model for High-Performance Extreme Multi-Label Text Classification. In *Advances in Neural Information Processing Systems*, Vol. 32. Curran Associates, Inc.

[206] Yousef Younes, Lukas Galke, and Ansgar Scherp. 2024. RADAr: A Transformer-based Autoregressive Decoder Architecture for Hierarchical Text Classification. In *European Conference on Artificial Intelligence*. AAAI.

[207] Chao Yu, Yi Shen, and Yue Mao. 2022. Constrained Sequence-to-Tree Generation for Hierarchical Text Classification. In *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022*, Enrique Amigó, Pablo Castells, Julio Gonzalo, Ben Carterette, J. Shane Culpepper, and Gabriella Kazai (Eds.). ACM, 1865–1869. https://doi.org/10.1145/3477495.3531765

[208] Chao Yu, Yi Shen, and Yue Mao. 2022. Constrained Sequence-to-Tree Generation for Hierarchical Text Classification. In *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022*, Enrique Amigó, Pablo Castells, Julio Gonzalo, Ben Carterette, J. Shane Culpepper, and Gabriella Kazai (Eds.). ACM, 1865–1869. https://doi.org/10.1145/3477495.3531765

[209] Hao Yu, Zachary Yang, Kellin Pelrine, Jean Francois Godbout, and Reihaneh Rabbany. 2023. Open, Closed, or Small Language Models for Text Classification? *arXiv:2308.10092* (2023).

[210] Linzhu Yu, Huan Li, Ke Chen, and Lidan Shou. 2024. BoKA: Bayesian Optimization based Knowledge Amalgamation for Multi-unknown-domain Text Classification. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2024, Barcelona, Spain, August 25-29, 2024*, Ricardo Baeza-Yates and Francesco Bonchi (Eds.). ACM, 4035–4046. https://doi.org/10.1145/3637528.3671963

[211] Zhizhi Yu, Di Jin, Jianguo Wei, Ziyang Liu, Yue Shang, Yun Xiao, Jiawei Han, and Lingfei Wu. 2022. TeKo: Text-Rich Graph Neural Networks with External Knowledge. *CoRR* abs/2206.07253 (2022). arXiv:2206.07253

[212] Zhiyi Yu, Hong Li, and Jialin Feng. 2023. Enhancing text classification with attention matrices based on BERT. *Expert Systems* (11 2023). https://doi.org/10.1111/exsy.13512

[213] Lifan Yuan, Yangyi Chen, Ganqu Cui, Hongcheng Gao, Fangyuan Zou, Xingyi Cheng, Heng Ji, Zhiyuan Liu, and Maosong Sun. 2023. Revisiting Out-of-distribution Robustness in NLP: Benchmark, Analysis, and LLMs Evaluations. *arXiv:2306.04618* (2023).

[214] Pingpeng Yuan, Yuqin Chen, Hai Jin, and Li Huang. 2008. MSVM-kNN: Combining SVM and k-NN for Multi-class Text Classification. In *IEEE International Workshop on Semantic Computing and Systems*. 133–140. https://doi.org/10.1109/WSCS.2008.36

[215] Alessandro Zangari, Matteo Marcuzzo, Matteo Rizzo, Lorenzo Giudice, Andrea Albarelli, and Andrea Gasparetto. 2024. Hierarchical Text Classification and Its Foundations: A Review of Current Research. *Electronics* 13, 7 (2024). https://doi.org/10.3390/electronics13071199

[216] Fang Zeng, Niannian Chen, Dan Yang, and Zhigang Meng. 2022. Simplified-Boosting Ensemble Convolutional Network for Text Classification. *Neural Process. Lett.* 54, 6 (2022), 4971–4986. https://doi.org/10.1007/s11063-022-10843-4

[217] Chaoqun Zhang, Linlin Dai, Chengxing Liu, and Longhao Zhang. 2025. HGBL: A Fine Granular Hierarchical Multi-Label Text Classification Model. *Neural Process. Lett.* 57, 1 (2025), 1. https://doi.org/10.1007/S11063-024-11713-X

[218] Dell Zhang, Jun Wang, Emine Yilmaz, Xiaoling Wang, and Yuxin Zhou. 2016. Bayesian Performance Comparison of Text Classifiers. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, SIGIR 2016, Pisa, Italy, July 17-21, 2016*. ACM, 15–24. https://doi.org/10.1145/2911451.2911547

[219] Jiong Zhang, Wei-Cheng Chang, Hsiang-Fu Yu, and Inderjit Dhillon. 2021. Fast Multi-Resolution Transformer Fine-tuning for Extreme Multi-label Text Classification. In *Advances in Neural Information Processing Systems*, Vol. 34. Curran Associates, Inc., 7267–7280.

[220] Jun Zhang, Yubin Li, Fanfan Shen, Chenxi Xia, Hai Tan, and Yanxiang He. 2024. Hierarchy-Aware and Label Balanced Model for Hierarchical Text Classification. *Knowl. Based Syst.* 300 (2024), 112153. https://doi.org/10.1016/J.KNOSYS.2024.112153

[221] Lu Zhang, Jiandong Ding, Yi Xu, Yingyao Liu, and Shuigeng Zhou. 2021. Weakly-supervised Text Classification Based on Keyword Graph. In *EMNLP (1)*. Association for Computational Linguistics, 2803–2813.

[222] Lu Zhang, Jiandong Ding, Yi Xu, Yingyao Liu, and Shuigeng Zhou. 2021. Weakly-supervised Text Classification Based on Keyword Graph. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*. Association for Computational Linguistics, 2803–2813. https://doi.org/10.18653/v1/2021.emnlp-main.222

[223] Min-Ling Zhang and Zhi-Hua Zhou. 2014. A Review on Multi-Label Learning Algorithms. *IEEE Transactions on Knowledge and Data Engineering* 26, 8 (2014), 1819–1837. https://doi.org/10.1109/TKDE.2013.39

[224] Wen Zhang, Yanbin Lu, Bella Dubrov, Zhi Xu, Shang Shang, and Emilio Maldonado. 2021. Deep Hierarchical Product Classification Based on Pre-Trained Multilingual Knowledge. *IEEE Data Eng. Bull.* 44, 2 (2021), 26–37. http://sites.computer.org/debull/A21june/p26.pdf

[225] Xinyi Zhang, Jiahao Xu, Charlie Soh, and Lihui Chen. 2020. LA-HCN: Label-based Attention for Hierarchical Multi-label Text Classification Neural Network. *CoRR* abs/2009.10938 (2020). arXiv:2009.10938

[226] Yu Zhang, Hao Cheng, Zhihong Shen, Xiaodong Liu, Ye-Yi Wang, and Jianfeng Gao. 2023. Pre-training Multi-task Contrastive Learning Models for Scientific Literature Understanding. *arXiv* (2023). arXiv:2305.14232 [cs.CL]

[227] Ye Zhang and Byron C. Wallace. 2017. A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing, IJCNLP 2017, Taipei, Taiwan, November 27 - December 1, 2017 - Volume 1: Long Papers*. Asian Federation of Natural Language Processing, 253–263. https://aclanthology.org/I17-1026/

[228] Yazhou Zhang, Mengyao Wang, Chenyu Ren, Qiuchi Li, Prayag Tiwari, Benyou Wang, and Jing Qin. 2024. Pushing The Limit of LLM Capacity for Text Classification. *CoRR* abs/2402.07470 (2024). https://doi.org/10.48550/ARXIV.2402.07470 arXiv:2402.07470

[229] Yufeng Zhang, Xueli Yu, Zeyu Cui, Shu Wu, Zhongzhen Wen, and Liang Wang. 2020. Every Document Owns Its Structure: Inductive Text Classification via Graph Neural Networks. In *The 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020*. ACL, 334–339. https://doi.org/10.18653/v1/2020.acl-main.31

[230] Ke Zhao, Lan Huang, Rui Song, Qiang Shen, and Hao Xu. 2021. A Sequential Graph Neural Network for Short Text Classification. *Algorithms* 14, 12 (2021), 352.

[231] Kaixin Zheng, Yaqing Wang, Quanming Yao, and Dejing Dou. 2022. Simplified Graph Learning for Inductive Short Text Classification. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*. Association for Computational Linguistics, 10717–10724. https://aclanthology.org/2022.emnlp-main.735

[232] Jie Zhou, Chunping Ma, Dingkun Long, Guangwei Xu, Ning Ding, Haoyu Zhang, Pengjun Xie, and Gongshen Liu. 2020. Hierarchy-Aware Global Model for Hierarchical Text Classification. In *Proceedings of the 58th Annual*

Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, Online, 1106–1117. https://doi.org/10.18653/v1/2020.acl-main.104

[233] Li Zhou, Wenyu Chen, Yong Cao, Dingyi Zeng, Wanlong Liu, and Hong Qu. 2024. MLPs Compass: What is learned when MLPs are combined with PLMs? *CoRR* abs/2401.01667 (2024). https://doi.org/10.48550/ARXIV.2401.01667 arXiv:2401.01667

[234] Pan Zhou, Jiashi Feng, Chao Ma, Caiming Xiong, Steven Chu Hong Hoi, and Weinan E. 2020. Towards Theoretically Understanding Why SGD Generalizes Better Than Adam in Deep Learning. In *Advances in Neural Information Processing Systems*, Vol. 33. Curran Associates, Inc., 21285–21296. https://proceedings.neurips.cc/paper/2020/file/f3f27a324736617f20abbf2ffd806f6d-Paper.pdf

[235] Peng Zhou, Zhenyu Qi, Suncong Zheng, Jiaming Xu, Hongyun Bao, and Bo Xu. 2016. Text Classification Improved by Integrating Bidirectional LSTM with Two-dimensional Max Pooling. In *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*. ACL, 3485–3495. https://aclanthology.org/C16-1329/

[236] Xujuan Zhou, Raj Gururajan, Yuefeng Li, Revathi Venkataraman, Xiaohui Tao, Ghazal Bargshady, Prabal Datta Barua, and Srinivas Kondalsamy-Chennakesavan. 2020. A survey on text classification and its applications. *Web Intell.* 18, 3 (2020), 205–216.

[237] He Zhu, Junran Wu, Ruomei Liu, Yue Hou, Ze Yuan, Shangzhe Li, Yicheng Pan, and Ke Xu. 2024. HILL: Hierarchy-aware Information Lossless Contrastive Learning for Hierarchical Text Classification. In *NAACL 2024*. ACL, 4731–4745. https://doi.org/10.18653/V1/2024.NAACL-LONG.265

[238] Yi Zhu, Ye Wang, Jipeng Qiang, and Xindong Wu. 2023. Prompt-Learning for Short Text Classification. *IEEE Transactions on Knowledge and Data Engineering* (2023), 1–13. https://doi.org/10.1109/TKDE.2023.3332787

[239] Daoming Zong and Shiliang Sun. 2022. BGNN-XML: Bilateral Graph Neural Networks for Extreme Multi-label Text Classification. *IEEE Transactions on Knowledge and Data Engineering* (2022), 1–12. https://doi.org/10.1109/TKDE.2022.3193657