Why Generate When You Can Discriminate?
A Novel Technique for Text Classification using Language Models

Sachin Pawar[1]*, **Nitin Ramrakhiyani**[1,2]*, **Anubhav Sinha**[1]
Manoj Apte[1], **Girish K. Palshikar**[1]
[1]TCS Research, Tata Consultancy Services Limited, India.
[2]International Institute of Information Technology (IIIT), Hyderabad, India.
{sachin7.p, nitin.ramrakhiyani, s.anubhav2, manoj.apte, gk.palshikar}@tcs.com

Abstract

In this paper, we propose a novel two-step technique for text classification using autoregressive Language Models (LM). In the first step, a set of perplexity and log-likelihood based numeric features are elicited from an LM for a text instance to be classified. Then, in the second step, a classifier based on these features is trained to predict the final label. The classifier used is usually a simple machine learning classifier like Support Vector Machine (SVM) or Logistic Regression (LR) and it is trained using a small set of training examples. We believe, our technique presents a whole new way of exploiting the available training instances, in addition to the existing ways like fine-tuning LMs or in-context learning. Our approach stands out by eliminating the need for parameter updates in LMs, as required in fine-tuning, and does not impose limitations on the number of training examples faced while building prompts for in-context learning. We evaluate our technique across 5 different datasets and compare with multiple competent baselines.

1 Introduction

In recent years, the autoregressive or causal language models (LM) such as GPT-3 (Brown et al., 2020) and GPT-Neo (Black et al., 2021) have been successful in a variety of natural language processing tasks such as summarization, machine translation, question answering, etc. Recently, there have been attempts to use such LMs for text classification (Min et al., 2022; Estienne, 2023; Sun et al., 2023) in a zero-shot or few-shot manner. In this paper, we propose a novel way of using moderate-sized (#parameters $\leq$ 2.7B) and open-source autoregressive language models for text classification. The central idea is that generating new text using LMs is not absolutely essential for text classification as is the case for other tasks such as summarization or machine translation, because the final goal is simply to discriminate among a finite set of class labels.

There are several challenges in using moderate-sized LMs like GPT-Neo-2.7B for text classification in both zero-shot as well as few-shot settings:

- In a zero-shot setting, getting the LM to generate an output containing the expected class labels is challenging. E.g., in case of the SST-2 (Socher et al., 2013) dataset for sentiment prediction, in spite of providing specific instruction in the prompt, for only around 10% test instances, the generated text contained the expected *Positive* and *Negative* labels. Most cases resulted in generation of some random text or text containing words like `mess` or `brilliant` from which inferring the actual labels is non-trivial (see Table 1).
- In a few-shot setting, the generated output conforms to the expected format in most cases. However, due to limited context window of the LM, a large number of training instances can not be provided in the prompt. This limits the ability of the LM to exploit a larger set of available labelled examples.
- Another way of exploiting training examples is through fine-tuning the LM. However, this requires specialized hardware resources (like GPUs with significant RAM) and time for fine-tuning.

Very large LMs like GPT-3 may not face these challenges, but their usage through API entails sharing the data to be classified and this may not be desirable for private and confidential data. Hence, in this paper, we focus on only moderate-sized LMs such as GPT-Neo-2.7B which can be deployed inhouse with very limited hardware. To overcome the above-mentioned challenges for such LMs, we propose a novel two-step technique for text classification. In the first step, for any text X to be classified, we elicit a set of feature values from the LM based on perplexity and log-likelihood of

[*] Equal contribution

Findings of the Association for Computational Linguistics: EACL 2024, pages 1099–1114
March 17-22, 2024 ©2024 Association for Computational Linguistics

Prompt: This is an overall sentiment classifier for movie reviews. Classify the overall SENTIMENT of the INPUT as Positive or Negative. INPUT: If this movie were a book, it would be a page-turner, you can't wait to see what happens next. SENTIMENT: *The movie is a mess.*
Prompt: This is an overall sentiment classifier for movie reviews. A review with Positive SENTIMENT finds the movie to be great, good, encouraging, brilliant, excellent, accurate, realistic, engaging, funny, or exciting. A review with Negative SENTIMENT finds the movie to be terrible, bad, unrealistic, frustrating, boring, forgettable, predictable, thoughtless, appalling, or incomprehensible. Classify the overall SENTIMENT of the INPUT as Positive or Negative. INPUT: Together, Tok and O orchestrate a buoyant, darkly funny dance of death. SENTIMENT: *Tok and O are a couple of misfits who...*

Table 1: Examples from SST-2 (sentiment prediction) through zero-shot text generation using GPT-Neo-2.7B. The generated text is shown in blue and *italics*.

certain label-specific augmentations of X. These augmentations are of the form "X. `This text is about ⟨key phrase⟩.`" where we simply need a set of *key phrases* associated with each class label. In a zero-shot setting, only this first step is required and a class label is predicted by a simple relative comparison of these feature values. In a supervised setting where labelled training instances are available, the second step is needed to train a light-weight machine learning classifier using the feature values obtained for the training instances. This classifier can then be used to predict the class label for any new instance to be classified.

The key phrases proposed in our approach are similar to the *verbalizers* used in techniques such as Pattern Exploiting Training (PET) (Schick and Schütze, 2021) and Knowledgeable Prompt-tuning (KPT) (Hu et al., 2022). However, these techniques are designed to work with encoder-only models like BERT (Devlin et al., 2019) or RoBERTa (Liu et al., 2019) whereas our technique is designed to work with decoder-only (causal) language models like GPT-2. A major limitation of techniques such as PET and KPT is that only single token verbalizers can be used for describing class labels. On the other hand, the key phrases used in our technique can be multi-word and hence overcome this major limitation. This is especially useful in real-life examples where multi-word key phrases are necessary, e.g., `fixed assets` (used in our experiments with financial audit reports in Section 6). Here, neither the individual words `fixed` and `assets` capture the complete underlying meaning nor a list of single token verbalizers (e.g., `land`, `machinery`) is sufficient enough. On the contrary, as our technique harnesses causal (decoder-only) models, it allows both single-word as well as multi-word key phrases. Moreover, techniques such as PET involve fine-tuning of the underlying model whereas our technique does not require such fine-tuning.

To summarize, the key contributions of this paper are as follows:

- A novel two-step technique for text classification using an autoregressive LM (Sections 3.2 and 3.3). Its key advantages are explainability and applicability in resource-poor settings as only inference using a moderate-sized LM is needed.
- Experimental evaluation to compare our technique with paradigms such as zero-shot prompting and few-shot in-context learning on topical as well non-topical text classification datasets (Section 5). Our technique is not restricted by the number of training instances, unlike in-context learning where the number of training instances are restricted by the LM's context length.
- Application to a real-life sentence classification problem in financial audit reports. (Section 6)

2 Perplexity and Log-likelihood

Perplexity is used as a metric to evaluate language models (Jurafsky and Martin, 2023). Intuitively, a better model of a text is the one which assigns a higher probability to a word that actually occurs. In this paper, we propose to use perplexity for a different purpose – judging *plausibility* of a text fragment using an autoregressive LM and comparing multiple such text fragments to decide which one is the most plausible. Here, by *plausibility* of a text, we mean that it is seemingly more reasonable or probable. A similar idea was explored by Lee et al. (2020) for detecting misinformation.

Consider a text fragment $X = [w_1, w_2, \cdots, w_n]$ which consists of n tokens. The perplexity of X as computed by an LM M is as follows:

$$PPL_M(X) = \prod_{i=1}^{n} \sqrt[n]{\frac{1}{P_M(w_i|w_{<i})}}$$

The *conditional perplexity* of a text fragment X given another text $C = [c_1, c_2, \cdots, c_m]$ as its prefix, can be computed as:

$$PPL_M(X|C) = \prod_{i=1}^{n} \sqrt[n]{\frac{1}{P_M(w_i|c_1, c_2, \cdots, c_m, w_{<i})}}$$

Similarly, *log-likelihood* and *conditional log-likelihood* for any text X are computed as follows:

$$LL_M(X) = \sum_{i=1}^{n} log(P_M(w_i|w_{<i}))$$
$$LL_M(X|C) = \sum_{i=1}^{n} log(P_M(w_i|c_1, \cdots, c_m, w_{<i}))$$

Overall, lower the perplexity of X (or higher the log-likelihood of X), better is its plausibility.

3 Text Classification

The task of text classification is to assign one or more applicable class labels from a pre-defined set of labels L to a piece of text X. There have been several attempts to use autoregressive LMs for text classification where a response is generated from an LM by providing the text to be classified as part of a prompt.

We hypothesize that there is no need to generate new text using an LM for text classification as we only need to discriminate among a finite set of class labels. Hence, rather than asking an LM to generate some new text, it is enough to simply compare plausibility of a set of text fragments (label-specific augmentations as shown in Table 2) where each augmentation corresponds to a specific class label. For the example sentence in Table 2, it can be clearly seen that out of all the label-specific augmentations, the texts A_{21} and A_{22} look comparatively more *plausible* and hence the corresponding class label Business is the most appropriate. Here, we expect that each class label is described by a set of *key phrases* based on the domain knowledge (examples in Table 2). There is no restriction on the number of key phrases to be used for each class, except that each class must have at least one key phrase which describes it. In absence of any domain knowledge, the class label itself can be used as one of the key phrases. For a more detailed discussion on key phrases, please refer Section 5.4. We now describe how we quantify the *plausibility* of these text fragments through multiple features (in Step 1) and learn a suitable function which maps these feature values to the appropriate class label (in Step 2).

3.1 Problem Setting

Input: (i) $L = \{L_1, \cdots, L_C\}$ (a set of C class labels), (ii) $P_i = \{p_1^i, \cdots, p_{n_i}^i\}$ (a set of n_i key phrases for each class label $L_i \in L$), (iii) $X = [w_1, w_2, \cdots, w_n]$ (text with n tokens to be classified), and (iv) M (an autoregressive LM)
Output: One or more class labels ($\subset L$) which are assigned to X
Training Regime: A small set of training instances where each instance is of the form $\langle X_t, L_t \rangle$ where L_t is a set of gold-standard labels for X_t such that $L_t \subseteq L$. In our experiments, we consider at most 500 training instances across all the datasets.

3.2 Step 1: Generating feature values

In this step, for each instance X (either text X to be classified or a training instance X_t), a set of feature values corresponding to each key phrase for each class label are obtained from the LM M. For each class label L_i, for its each key phrase p_j^i, the following two feature values are obtained.

$$f_{ij}^{PPL}(X) = \frac{PPL_M(p_j^i|X+S)}{PPL_M(p_j^i|S)}$$
$$f_{ij}^{LL}(X) = LL_M(p_j^i|X+S) - LL_M(p_j^i|S)$$

Here, the first feature captures reduction in perplexity of the key phrase p_j^i and the second feature captures increase in its log-likelihood, when X is provided as part of its prefix. Although there is inter-dependence between perplexity and log-likelihood, considering both PPL and LL features is necessary and a detailed discussion is presented in Appendix A.3.

To ensure a proper English sentence formation which links the key phrase to its prefix X, we use a connector sentence S of the form `This news is about`[1]. So, $X + S$ forms the prefix context of a key phrase as shown in Table 2. The intuition is that if the key phrase p_j^i is semantically related to the text X, its conditional perplexity $PPL_M(p_j^i|X + S)$ when conditioned on $X + S$ should be lower than $PPL_M(p_j^i|S)$ which is only conditioned on S. Hence, lower the $f_{ij}^{PPL}(X)$ value, higher the chance that the text is really about p_j^i. Similarly, higher the $f_{ij}^{LL}(X)$ value, higher the chance that the text is about p_j^i. For the example sentence in Table 2, these feature values are shown for various key phrases. Also, the choice of a connector sentence does not have much effect on the final predictions because – (i) S is common across all the key phrases for a given dataset and (ii) S is conditioned upon in both the

[1] We use different connector sentences for different datasets as shown in Section 5.1.

Text to be classified, X = Expansion slows in Japan. Economic growth in Japan slows down as the country experiences a drop in domestic and corporate spending.	**Class labels with corresponding key phrases:** Sports: sports, a sporting event, a sportsperson, ... Business: business, economy, stock market, ... Science: science, space exploration, software, ...

Label-specific augmentations of the above sentence	f_{ij}^{PPL}	f_{ij}^{LL}
A_{11}: Expansion slows in Japan. Economic growth in Japan slows down as the country experiences a drop in domestic and corporate spending. This news is about **sports**.	3.48	-2.50
A_{12}: Expansion slows in Japan. Economic growth in Japan slows down as the country experiences a drop in domestic and corporate spending. This news is about **a sporting event**.	1.42	-1.42
A_{21}: Expansion slows in Japan. Economic growth in Japan slows down as the country experiences a drop in domestic and corporate spending. This news is about **business**.	1.22	-0.40
A_{22}: Economic growth in Japan slows down as the country experiences a drop in domestic and corporate spending. This news is about **economy**.	0.62	0.95
A_{31}: Expansion slows in Japan. Economic growth in Japan slows down as the country experiences a drop in domestic and corporate spending. This news is about **science**.	7.12	-3.92
A_{32}: Expansion slows in Japan. Economic growth in Japan slows down as the country experiences a drop in domestic and corporate spending. This news is about **space exploration**.	1.52	-1.27

Table 2: Illustration of our text classification approach. In each label-specific augmentation, the text to be classified (X) is shown in black, the connector sentence (S) is shown in brown and the key phrases are shown in blue. The f_{ij}^{PPL} and f_{ij}^{LL} feature values are computed using the GPT2-XL model.

terms $PPL_M(p^i_j|X+S)$ and $PPL_M(p^i_j|S)$ (also $LL_M(p^i_j|X+S)$ and $LL_M(p^i_j|S)$) and hence the effect of any specific S is cancelled. We empirically observed this in our experiments in Figure 3. The only purpose of S is to construct a well formed and suitable English sentence which connects the key phrase with X as its prefix.

In addition to the above *keyphrase-level* features, for each class label L_i, two *class-level* features are added as follows:

$$f_i^{PPL}(X) = min_j \left(f_{ij}^{PPL}(X)\right)$$
$$f_i^{LL}(X) = max_j \left(f_{ij}^{LL}(X)\right)$$

Intuitively, for each class, the best feature values across all its key phrases are stored as separate class-level features. Hence, overall for each instance X, the number of features is equal to $2 \cdot \left(\sum_{i=1}^{C}(n_i) + C\right)$.

Zero-shot classification (ZS-PPL/ZS-LL): The above feature values computed for any text X are themselves enough to predict a class label in zero-shot manner. Here, the predicted class label is the one whose key phrase led to the minimum perplexity ratio or the maximum log-likelihood increase.

$$ZS\text{-}PPL(X) = argmin_i \left(f_i^{PPL}(X)\right)$$
$$ZS\text{-}LL(X) = argmax_i \left(f_i^{LL}(X)\right)$$

3.3 Step 2: Learning a classifier

This step is needed only in case of a supervised setting where labelled training instances are available. In the above zero-shot classification rule (ZS^{PPL}/ZS^{LL}), a very simple function which maps the feature values to a class label is used, i.e., simply considering *minimum* or *maximum* over certain feature values. On the other hand, if training instances are available, a more complex function which maps these feature values to a class label can be learned. As one of the ways to learn such a function, in this step, we simply learn a supervised machine learning classifier using the feature values obtained for the training instances. This classifier can then be used to predict class labels for new unseen instances. We explored multiple lightweight classifiers and observed logistic regression (LR) and support vector machines (SVM) to be the best performing in both multi-class and multi-label (one-vs-all) settings.

3.3.1 Horizontal Scaling

We scaled the feature values for each instance such that minimum feature value is set to 0 and the maximum is set to 1. We did such scaling separately for perplexity based features and log-likelihood based features. Please note that this is different from the usual min-max scaling[2] where a fixed feature is scaled across multiple instances, whereas we are scaling multiple features for a fixed instance. Intuitively, our feature values are such that the comparison of relative values of these features with each other is important for determining the final class label.

[2] https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html

Discussion on explainability: The predictions of the proposed technique are explainable by design. For each predicted label, an explanation is generated in the form of a ranked list of key phrases (sorted using f_{ij}^{PPL} or f_{ij}^{LL}) associated with the predicted class (examples in Table 10).

4 Related Work

While LMs enhance performance across various NLP tasks, prior research has revealed several challenges when applying them to text classification, such as designing appropriate prompts in zero-shot setting, limited input prompt length when using in-context learning, and costly as well as time-consuming fine-tuning. Given these constraints, there is a line of research which explores novel ways using moderate-sized LMs for text classification. One of the recent prominent work in this area is by Min et al. (2022). They introduce "noisy channel" as well as "direct" methods which compute conditional probability of the input text given the label or vice versa, for few-shot text classification through in-context learning and prompt tuning. Our proposed technique resembles their approach to some extent in computing conditional perplexity, but there are several key differences – (i) computing multiple features using domain knowledge based key phrases, (ii) no limitation on number of training examples, and (iii) learning a classifier based on these features.

Another relevant work for our technique is by Estienne (2023) wherein the authors propose to calibrate output probabilities of an LM through prior adaptation to perform text classification tasks. They propose two variations of their approach – unsupervised (UCPA) where no labelled data is needed and semi-unsupervised (SUCPA) where some training examples (600) are used for prior adaptation. Both Min et al. (2022) and Estienne (2023) are most relevant for our technique in the sense that they only use moderate-sized LMs such as GPT2-XL and hence we consider both of these as important baselines.

A recent approach by Sun et al. (2023) presents an innovative approach by integrating the general language understanding of LLMs with task-specific data in the form of clues and reasoning from labeled datasets, providing an effective solution. Another work by Hou et al. (2023) focuses on a method for building a text classifier from an LLM all within a *black box* paradigm, without direct access to internal model parameters. Yang and Liu (2022) introduces a robust prefix-tuning framework, enhancing robustness while maintaining efficiency, particularly in the context of text classification. This is achieved by leveraging language model activation and batch-level prefix tuning.

Dataset	#instances		#labels	#key phrases
	train	test		
SST-2	$500^\dagger$	1821	2	20
TREC	$500^\dagger$	500	6	50
AGNews	$500^\dagger$	7600	4	37
DBPedia	$500^\dagger$	$1000^\dagger$	14	41
Ethos	$200^\dagger$	$233^\dagger$	8^*	20

Table 3: Dataset Details. † indicates the randomly chosen instances from the original train/test split whereas other numbers are original test split. * indicates multi-label setting.

Meng et al. (2022) presented an interesting technique where a causal LM generates class-conditioned texts guided by prompts, which are used as the training data for fine-tuning an encoder-only model. We believe that auto-generating new training instances is reasonable for simpler text classification problems like SST2 but not for TREC (Section 5.1) which is a more challenging text classification problem. In TREC, because the text to be classified is a question and the expected label is its answer type, it is not trivial to come up with a answer type based prompt which can generate suitable questions as expected in the technique by Meng et al. (2022). Our CHT-BERT baseline (Section 5.2) is similar where labelled instances are used for fine-tuning the encoder model. In fact, this baseline is more competitive than Meng et al. (2022) given it uses gold-standard labelled instances instead of auto-generated instances.

5 Experiments

5.1 Datasets

We use 5 datasets with different properties for all our experiments. Broadly, the text classification task is of two types – (i) *topical* where the class labels roughly correspond to the *topics* being discussed in the text and (ii) *non-topical* where the class labels generally correspond to some semantic property of the text as a whole. We consider two popular *topical* datasets – AGNews (Zhang et al., 2015) (4 classes) and DBPedia (Lehmann et al., 2015) (14 classes). We also consider two popular *non-topical* datasets – SST-2 (Socher et al., 2013) which is a binary sentiment analysis dataset and

	SST-2	TREC	AGNews	DBPedia	Ethos
Baselines:					
ZS-KP (zero-shot with keyphrases)	0.248	0.020	0.039	0.182	0.035
ZS-KP-CoT (ZS-KP with Chain-of-Thought)	0.061	0.046	0.024	0.239	0.019
FS-ICL	0.814	0.308	0.672	0.689	0.438
CHT	0.620	0.734	0.691	0.558	0.164
Our proposed techniques:					
ZS-PPL (zero-shot with only PPL features)	0.752	0.384	0.787	0.735	0.527
ZS-LL (zero-shot with only LL features)	0.766	0.418	0.774	0.67	0.438
SVM with all features and horizontal scaling	**0.893**	**0.804**	**0.860**	0.912	0.671
LR with all features and horizontal scaling	**0.893**	0.798	0.858	**0.926**	**0.673**

Table 4: Comparison of baselines and proposed approach for the GPT-Neo-2.7B model.

	SST-2	TREC	AGNews	DBPedia	Ethos
Unsupervised Calibration through Prior Adaptation (Estienne, 2023)					
SUCPA (zero-shot)	0.850	0.460	0.700	0.660	NA
SUCPA (few-shot)	0.890	0.550	0.780	0.880	NA
Noisy Channel Language Model Prompting[†] (Min et al., 2022)					
Channel (zero-shot)	0.771	0.305	0.618	0.514	NA
Channel (concat-based)	0.850	0.420	0.685	0.585	NA
Channel (ensemble-based)	0.775	0.315	0.743	0.648	NA
Other baselines:					
ZS-KP (zero-shot with keyphrases)	0.183	0.10	0.088	0.157	0.137
ZS-KP-CoT	0.160	0.01	0.029	0.089	0.032
FS-ICL	0.874	0.476	0.330	0.085	0.182
CHT	0.567	0.476	0.592	0.488	0.029
CHT-BERT*	0.890	0.698	0.801	0.834	0.219
Our proposed techniques:					
ZS-PPL (zero-shot with only PPL features)	0.871	0.478	0.776	0.762	0.479
ZS-LL (zero-shot with only LL features)	0.875	0.462	0.764	0.716	0.421
SVM with all features and horizontal scaling	0.919	**0.860**	0.851	0.912	0.707
LR with all features and horizontal scaling	**0.920**	0.824	**0.853**	**0.924**	**0.715**

Table 5: Comparison of baselines and proposed approach for the GPT2-XL model. ([†]These numbers are using GPT2-Large model and the authors have observed similar performance for GPT2-XL making it comparable. *The baseline CHT-BERT is based on the encoder model bert-large-uncased.)

TREC (Voorhees and Tice, 2000) where one of the 6 answer types are to be predicted for various questions. In addition to these single-label datasets, we also consider a multi-label dataset Ethos (Mollas et al., 2020) where the goal is to predict one or more hate types for a hate speech comment. The details about all the datasets are shown in Table 3. Table 9 shows the set of key phrases used for each class in these datasets. The connector sentences used for the different datasets are as follows:

- SST2: `This comment finds the movie to be`
- TREC: `The answer will be`
- AGNews: `This news is about`
- DBPedia: `This text is about`
- Ethos: `This comment is about`

5.2 Baselines

ZS-KP: As a variant of the vanilla zero-shot prompting approach, which guides the LM only based on the instruction for the task, we use a zero-shot with key phrases baseline. Along with the task instruction, we include the definition of the class label in terms of the key phrases which we use in the proposed approach. One sentence per class label is added to the prompt followed by the task instruction. E.g., to explain the AGNews' `Sports` class, we add the sentence `The Sports TOPIC news is about sports, a sporting event, sporting awards, a sports champion, a sportsperson, wins or losses in sports, or prize money.` to the prompt (a similar example for SST2 is shown in Table 1).

ZS-KP-CoT: This is a variant of the above ZS-KP baseline which also includes a Chain-of-Thought (CoT) instruction to press the LM to arrive at the answer, reasoning through a step-by-step process. We append the instruction *Let's think step-by-step.* as proposed in (Kojima et al., 2022) to the prompt in ZS-KP and parse the output to arrive at the predicted class label. We evaluate the predictions for both ZS-KP and ZS-KP-CoT leniently, where we consider the prediction to be correct even if the exact class name is not present in the generated text, but a corresponding key phrase is.

FS-ICL: As part of the few shot in-context learning (Brown et al., 2020) baseline, we randomly select a set of k (= 16) examples from the training data and build a prompt with the instruction and selected examples. Finally, we append the input test instance and obtain the class label. In this FS-ICL baseline, the LMs considered were able to predict the exact class label and did not require any answer parsing as in the above zero-shot baselines.

CHT: We also consider a supervised baseline, where we tune a classification head (CH) on top of the LM using the exactly same labelled examples we consider for training our classifiers in Step 2. However, we do not allow the layers of the LM to get trained thereby keeping its inherent pre-training intact. This baseline gives the necessary comparison with the proposed technique where labelled examples are used without fine-tuning the LM.

5.3 Results and Analysis

For all our experiments, we considered two moderate-sized autoregressive LMs – GPT-Neo-2.7B (Black et al., 2021) and GPT2-XL (Radford et al., 2019). The focus of our experiments was to compare multiple techniques of using the same model for text classification. For all datasets except Ethos, the *accuracy* is used as the evaluation metric whereas for the multi-label Ethos dataset, *micro-averaged F1-score* across class labels is used.

Table 4 shows the experimental results for the GPT-Neo-2.7B model. Here, our techniques - SVM and LR classifiers, are outperforming all other baselines. Even our zero-shot technique ZS-PPL, outperforms the few-shot baseline for TREC, AG-News, DBPedia and Ethos. Table 5 shows the experimental results for the GPT2-XL model. The reason for choosing this model for experiments was mainly to compare our results with Estienne (2023) which is the most relevant prior work. In case of GPT2-XL model as well, our techniques are outperforming all other baselines, including Estienne (2023). Again, our zero-shot techniques ZS-PPL and ZS-LL, outperform the few-shot baseline for AGNews, DBPedia and Ethos. ZS-PPL and ZS-LL also outperform the channel models of Min et al. (2022) in both zero-shot as well as few-shot settings. We also experimented with another baseline CHT-BERT, a variant of CHT using an encoder-only model (bert-large-uncased). Though CHT-BERT outperforms CHT, our supervised technique proves to be better than this CHT-BERT baseline.

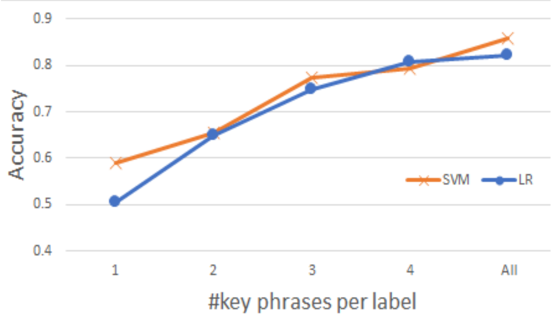

Figure 1: Accuracy for TREC with varying number of key phrases per class label using GPT2-XL model

Overall, our technique focuses on improving performance as compared to the standard prompting techniques for moderate-sized causal LMs which we prefer to use because they are open source and easy to deploy with moderate hardware. Hence, we are not achieving SOTA results achieved by larger models (Table 11) or encoder models (Hu et al., 2022). We feel that a fair comparison would be with techniques using similar sized causal LMs (e.g., GPT2-XL). Hence, we have added two such baselines based on the recent work (Min et al., 2022; Estienne, 2023). Further, we would like to highlight that our technique can be generalized to different types of text classification problems (*non-topical* as well as *topical*) which is evident from our results (Table 4 and 5) on 5 text classification datasets of different nature.

Ablation Analysis: We carried out detailed ablation analysis to quantify the contribution of each of the following – (i) horizontal scaling, (ii) perplexity-based (PPL) features, (iii) log-likelihood-based (LL) features, (iv) keyphrase-level features, and (v) class-level features. Table 6 shows the ablation analysis results for the GPT2-XL model. Horizontal scaling is clearly observed to be useful across all the datasets, because the performance degrades without such scaling. Similarly, LL features and keyphrase-level features are observed to be useful consistently across all the datasets. The class-level features are also similarly observed to be useful, though the decrease in accuracy is not prominent. On the other hand, mixed results are observed for the PPL features across multiple datasets for the GPT2-XL model.

Effect of number of key phrases: To measure the contribution of using multiple key phrases, we carried out two experiments. The first experiment evaluates performance of our classifiers in the ex-

	SST-2	TREC	AGNews	DBPedia	Ethos
SVM default setting: With all features and horizontal scaling	0.919	**0.860**	0.851	**0.912**	0.707
SVM default setting without Horizontal scaling	0.902	0.814	0.768	0.911	0.653
SVM default setting without LL features	0.916	0.648	0.825	0.888	0.639
SVM default setting without PPL features	0.916	0.840	**0.855**	0.909	**0.710**
SVM default setting without class-level features	**0.921**	0.858	0.845	0.907	0.707
SVM default setting without keyphrase-level features	0.869	0.576	0.781	0.896	0.673
SVM default setting with only one keyphrase per class	0.832	0.590	0.684	0.856	0.660
LR default setting: With all features and horizontal scaling	**0.920**	**0.824**	0.853	**0.924**	**0.715**
LR default setting without Horizontal scaling	0.908	0.820	0.792	0.911	0.686
LR default setting without LL features	0.914	0.684	0.828	0.884	0.633
LR default setting without PPL features	0.919	**0.824**	**0.856**	0.916	0.712
LR default setting without class-level features	0.918	0.822	0.850	0.917	0.703
LR default setting without keyphrase-level features	0.880	0.486	0.784	0.886	0.672
LR default setting with only one keyphrase per class	0.832	0.504	0.688	0.855	0.647

Table 6: Ablation analysis with the GPT2-XL model (see Table 12 for the GPT-Neo-2.7B model)

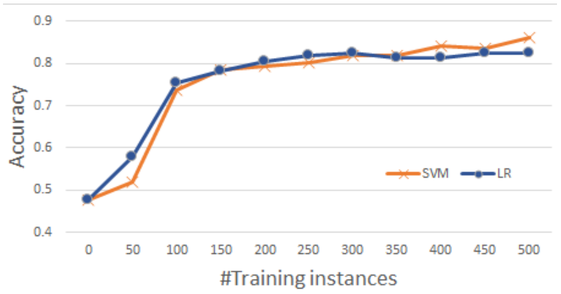

Figure 2: Accuracy for TREC with varying number of training instances per class label using GPT2-XL model

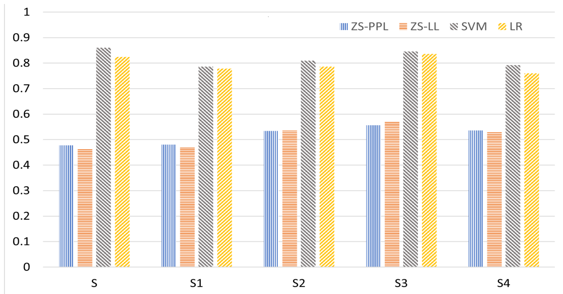

Figure 3: Accuracy for TREC with various connector sentences using GPT2-XL (S:`The answer will be`, S1: `The answer will be about`, S2: `The answer is`, S3:`The answer must be`, S4: `The answer is about`)

treme case of using just one key phrase per class. The last rows for SVM and LR in Table 6 shows the accuracy numbers for all datasets in this case (we used the first key phrase for each class in Table 9). Even though there is a significant drop in accuracy as compared with the default setting, the accuracy is still better than the few-shot and CHT baselines for most of the datasets. The second experiment evaluates the effect of varying the number of key phrases used per class for the TREC dataset as shown in Figure 1. With just 4 key phrases per class, accuracy close to 0.8 is observed.

Effect of number of training instances: We evaluated the effect of varying the number of training instances for the TREC dataset as it had the largest difference between the zero-shot and supervised (SVM/LR) accuracy. Figure 2 shows the accuracy when the number of training instances are increased from 50 to 500. There is a sharp increase till around 200 instances after which it gets plateaued.

Effect of different connector sentences: We also evaluated the effect of using multiple connector sentences for TREC as shown in Figure 3 where S is our default connector. Though a small difference is observed in accuracy, even the worst case accuracy for SVM (0.786) is better than all other baselines for TREC using GPT2-XL.

5.4 Discussion on acquisition of key phrases

For some classification problems, obtaining key phrases would be non-trivial and may require some domain knowledge. However, in complex real-life classification problems, it might be easier and faster to obtain key phrases from domain experts or documented domain knowledge than to get sufficient annotations from them. We experienced this in our analysis of financial audits (Section 6). In this case, the existing domain knowledge was available as part of standard auditing checklists and guidelines, which were used to obtain initial set of key phrases with minimum efforts. Also, another example would be of the TREC dataset where we have simply used fine-grained labels (already provided as part of the dataset/task) as the key phrases for the 6 coarse-grained labels. In all our experiments, we have used at the most 10 key phrases per class label. And in most cases, the number of key phrases per class is less than that (Tables 9 and 15).

#training instances	SVM	LR	ZS-PPL	ZS-LL	CG
1097	0.542	0.536	0.380	0.410	0.520
500	0.503	0.498			

Table 7: Performance on Audit Reports test dataset

Hence, we believe for any classification problem, it would be reasonable to assume that such small set of key phrases can be identified without any major difficulty, either from domain experts, documented domain knowledge, or from any other relevant knowledge bases.

6 Analysis of Financial Audit Reports

Financial audit is a complex process used by organizations to assure the stakeholders about the quality and trustworthiness of the governance (Whittington and Pany, 2021; Arens and Loebbecke, 1999). One important outcome of an audit is the *audit report*, wherein the auditor declares the financial statements of a company are free from material misstatement, are fair and accurate and are presented in accordance with the relevant accounting standards. A good comprehensive audit report is an important indicator of a good audit. Audit monitoring bodies such as The Chartered Accountants (CA) Society of India have issued guidelines on the contents of audit reports wherein they describe a set of audit aspects which the auditor should touch upon and describe. The problem of verifying whether an audit report has covered these audit aspects, can be modelled as a multi-class multi-label text classification problem where each sentence in the report can be labelled with zero or more audit aspects. We have identified a set of 15 audit aspects from standard auditing checklist (ICAI, 2017) and Companies (Auditor's Report) Order, 2020 (CARO) (ICAI, 2020), such as payables, inventory, and fixed assets (see Table 14 for complete list).

Audit Dataset: We used the 3744 web-scraped audit reports made available by Maka et al. (2020) for the year 2014. As getting *gold-standard* labelled examples was time and effort intensive, we automatically obtained *silver-standard* training data (1097 sentences) with the help of regular expression based patterns. These patterns were constructed using a set of key phrases obtained for each class by consulting domain experts (Table 15). We used the same set of key phrases in our technique for this classification problem.

Test dataset: For evaluating the classification performance, a set of 10 audit reports (1668 sentences) were labelled manually by domain experts.

Results: Table 7 shows the micro-averaged F1-scores on the test dataset, using GPT2-XL. We also compare with a ChatGPT baseline using zero-shot prompting (full prompt in Table 13) and observe a comparable performance.

To summarize, this was a challenging multi-label classification problem with no labelled sentences available. With the help of the proposed technique, we were able to quickly build a classification system which – (i) captures domain knowledge about audit aspects in terms of multiple corresponding key phrases, (ii) can be deployed in-house with limited resources to avoid sharing the data outside the organization, (iii) provides some explanations with each predicted label, and (iv) achieves reasonable performance (comparable with zero-shot ChatGPT) with a moderate-sized open-source LM, though there is still scope for improvement.

7 Conclusions and Future Work

We proposed a novel two-step technique for text classification using moderate-sized (#params $\leq$ 2.7B) autoregressive Language Models (LM). In the first step, for a text instance to be classified, a set of perplexity and log-likelihood based features are obtained from an LM. A light-weight classifier (SVM or LR) is trained in the second step to predict the final label. Our technique presents a new way of exploiting the available labelled instances, in addition to the existing ways such as fine-tuning LMs or in-context learning. It neither needs any parameter updates in LMs as in fine-tuning nor it is restricted by the number of training examples to be provided in the prompt for in-context learning. The key advantages of our technique are its explainability through most suitable key phrases and its applicability in resource poor environments. We demonstrate effectiveness of the proposed technique by comparing it with multiple baselines in the context of two LMs (GPT-Neo-2.7B and GPT2-XL) on five different datasets.

In future, we plan to extend this work by – (i) automatically discovering optimal set of key phrases and connector sentences, (ii) learning a function which exploits the inter-dependence between multiple features in a better way, (iii) exploring an ensemble where features from multiple LMs are combined, and (iv) evaluating the generated explanations quantitatively through a user study.

8 Limitations

Some key limitations of our proposed technique are as follows:

- Our approach needs a set of key phrases for each class label. Generally, these should be available (such as in case of TREC where we simply used the fine-grained labels as key phrases for corresponding coarse-grained labels) or can be constructed easily (as very few key phrases are required) for general domain classification problem. Though, in some domain-specific classification problems, availability of domain experts would be must. As of now, automatically discovering an optimal set of key phrases as well as connector sentences, is not tackled.
- The current work does not explore whether the proposed idea also works well with larger LMs (#params >> 2.7B such as Falcon-40B, GPT-3) where text generation capabilities are much better. For example, Table 11 shows that techniques based on GPT-3 text generation, lead to better performance as compared with our technique based on much smaller models.
- As of now, we have used perplexity (and log-likelihood) based features for a specific label-specific augmentations of text to be classified. However, the current work does not explore other forms of such augmentations.
- We have randomly sampled 500 training examples for each dataset just once. The purpose of the experiment was to compare our technique with the CHT baseline and we use exactly the same set of 500 training examples for training in CHT as well.

References

Alvin A. Arens and James K. Loebbecke. 1999. *Auditing: An Integrated Approach*, 8th edition. Pearson.

Sid Black, Gao Leo, Phil Wang, Connor Leahy, and Stella Biderman. 2021. GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Lautaro Estienne. 2023. Unsupervised calibration through prior adaptation for text classification using large language models. *arXiv preprint arXiv:2307.06713*.

Bairu Hou, Joe O'connor, Jacob Andreas, Shiyu Chang, and Yang Zhang. 2023. Promptboosting: Black-box text classification with ten forward passes. In *International Conference on Machine Learning*, pages 13309–13324. PMLR.

Shengding Hu, Ning Ding, Huadong Wang, Zhiyuan Liu, Jingang Wang, Juanzi Li, Wei Wu, and Maosong Sun. 2022. Knowledgeable prompt-tuning: Incorporating knowledge into prompt verbalizer for text classification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2225–2240, Dublin, Ireland. Association for Computational Linguistics.

ICAI. 2017. Internal audit checklist. `https://kb.icai.org/pdfs/44970iasb34918.pdf`. [Online; accessed 8-September-2023].

ICAI. 2020. ICAI'S GUIDANCE NOTE ON CARO 2020 (CARO). `https://wirc-icai.org/wirc-reference-manual/part2/icai-guidance-note-on-caro-2020.html`. [Online; accessed 8-September-2023].

Dan Jurafsky and James H. Martin. 2023. *Speech and Language Processing, 3rd edition*. Online version accessed on 20-SEP-2023.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Nayeon Lee, Yejin Bang, Andrea Madotto, and Pascale Fung. 2020. Misinformation has high perplexity. *arXiv preprint arXiv:2006.04666*.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. 2015. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic web*, 6(2):167–195.

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. *arXiv preprint arXiv:1907.11692*.

Kiran Maka, S. Pazhanirajan, and Sujata Mallapur. 2020. Selection of most significant variables to detect fraud in financial statements. *Materials Today: Proceedings*.

Yu Meng, Jiaxin Huang, Yu Zhang, and Jiawei Han. 2022. Generating training data with language models: Towards zero-shot language understanding. *Advances in Neural Information Processing Systems*, 35:462–477.

Sewon Min, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Noisy channel language model prompting for few-shot text classification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5316–5330.

Ioannis Mollas, Zoe Chrysopoulou, Stamatis Karlos, and Grigorios Tsoumakas. 2020. Ethos: an online hate speech detection dataset.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Timo Schick and Hinrich Schütze. 2021. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269, Online. Association for Computational Linguistics.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Xiaofei Sun, Xiaoya Li, Jiwei Li, Fei Wu, Shangwei Guo, Tianwei Zhang, and Guoyin Wang. 2023. Text classification via large language models. *arXiv preprint arXiv:2305.08377*.

Ellen M Voorhees and Dawn M Tice. 2000. Building a question answering test collection. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 200–207.

Ray Whittington and Kurt Pany. 2021. *Principles of Auditing and Other Assurance Services*, 22 edition. McGraw-Hill Education.

Zonghan Yang and Yang Liu. 2022. On robust prefix-tuning for text classification.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.

A Additional Details

A.1 Key phrases

Table 9 shows the key phrases used for each class label in the SST-2, AGNews, TREC, DBPedia, and Ethos datasets. Specifically for the TREC dataset, as we are using only 6 coarse labels, we use the 50 fine-grained labels as the corresponding key phrases.

$log(p(w_1))$	$log(p(w_2))$	$log(p(w_3))$	PPL	LL
-1.8	-2.5	NA	8.58	-4.3
-1.1	-2.1	-2.0	5.66	-5.2

Table 8: Example showing differing relative orderings of PPL and LL values

A.2 Examples of explanations

Table 10 shows the explanations for the predicted labels in terms of the key phrases corresponding to the minimum value of f_{ij}^{PPL} for each instance.

A.3 Discussion on dependence between PPL and LL

As we know, perplexity and log-likelihood are related as follows: $PPL_M(p) = \exp\left(\frac{-1}{n} LL_M(p)\right)$ where n is the number of tokens (word pieces) within p. This would imply that when the key phrases consist of exactly the same number of tokens (n), then we would obtain exactly the same ordering of the feature values for both PPL and LL based features. This would in-turn lead to the same predictions by both ZS-PPL and ZS-LL. But in practice, the key phrases may contain different number of tokens, leading to different relative ordering of PPL and LL based features. As can be seen in the example in Table 8 where the first key phrase (having 2 tokens) has a better LL than the second key phrase (having 3 tokens) but vice versa in case of PPL. Hence, exploring both PPL and LL based features is important.

A.4 Implementation Details

Perplexity and Log-likelihood: We used the HuggingFace transformers library[3] for computing perplexity and log-likelihood values using the models GPT-Neo-2.7B[4] and GPT2-XL[5]. The negative log-likelihood loss values returned by the models GPTNeoForCausalLM and GPT2LMHeadModel

[3] https://huggingface.co/

[4] https://huggingface.co/EleutherAI/gpt-neo-2.7B

[5] https://huggingface.co/gpt2-xl

Dataset	Label	Key phrases
SST-2	Positive	great, good, encouraging, brilliant, excellent, accurate, realistic, engaging, funny, exciting
	Negative	terrible, bad, unrealistic, frustrating, boring, forgettable, predictable, thoughtless, appalling, incomprehensible
AGNews	World	politics, terrorism, president of a country, a military related event, minister of a country, elections and government formation, a natural disaster, a war or an armed conflict, protests or demonstration, religious events
	Sports	sports, a sporting event, sporting awards, a sports champion, a sportsperson, wins or losses in sports, prize money
	Business	business, stock market, banking, monetary investments, economy, income and expenditure, corporate profit and loss, international trade, sale of goods and services, monetary policies
	Science	science, technology and engineering, research and development, internet and web, space exploration, cyber security, software, weather and climate, healthcare and pharma, flora and fauna
TREC	ABBR	an abbreviation, an expression which is abbreviated
	ENTY	an entity, an animal, an organ of body, a color, an invention, book and other creative piece, a currency name, a disease or a medicine, an event, food, a musical instrument, a language, a letter or a character, a plant, a product, a religion, a sport , a chemical element or a substance ,a symbol or a sign, a technique or a method, an equivalent term, a vehicle, a word with a special property
	DESC	description of something, a definition of something, a manner of an action, a reason
	HUM	an individual, a group or organization of persons, a title of a person, description of a person
	LOC	a location, a country, a mountain, a city, a state
	NUM	a number, a postcode or other code, number of something, a date, distance or linear measure, price, order or rank, period or lasting time of something, percent or fraction, speed, temperature, size, area or volume, weight
DBPedia	Company	a company, an organization
	EducationalInstitution	an educational institution, a school, a college
	Artist	an artist, a painter, a singer, a musician, an actor, an entertainer, a scientist
	Athlete	an athelete, a sportsperson
	OfficeHolder	a designation held by someone, a politician, a lawmaker
	MeanOfTransportation	a vehicle, a car, a train, an aeroplane, a ship or boat
	Building	a building, a monument, a man-made structure
	NaturalPlace	a natural location, a natural reserve
	Village	a village, a town
	Animal	an animal species, an insect, a bird, a fish, a reptile
	Plant	a plant species
	Album	an album
	Film	a film, a movie
	WrittenWork	a book, a magazine, a novel
Ethos	violence	violence, physically hurting someone
	directed_vs_generalized	specific individual as target
	gender	gender, women
	race	race, white people, black people
	national_origin	national origin, people from a specific country
	disability	disability, people with specific disorder or disability
	religion	religion, Islam, Christianity, Judaism, Hinduism
	sexual_orientation	sexual orientation, transgenders, homosexuality

Table 9: Key phrases used in all the datasets

Text	Label	Key phrase	f_{ij}^{PPL}
Afghan Army Dispatched to Calm Violence. KABUL, Afghanistan - Government troops intervened in Afghanistan's latest outbreak of deadly fighting between warlords, flying from the capital to the far west on U.S. and NATO airplanes to retake an air base contested in the violence, officials said Sunday...	World	terrorism	0.259
Late rally sees Wall Street end week on a positive note. US BLUE-chips recovered from an early fall to end higher as a drop in oil prices offset a profit warning from aluminium maker Alcoa, while a rise in Oracle fuelled a rally in technology stocks after a judge rejected a government attempt to block a...	Business	stock market	0.087
Bekele, Isinbayeva top track athletes. Names Ethiopian distance runner Kenenisa Bekele and Russian pole vaulter Yelena Isinbayeva were named male and female athletes of the year by the world track and field federation. Isinbayeva set eight world records in 2004, including one while winning the gold medal at the Olympics. Bekele won the 10,000 meters in Athens and finished second to Hicham El Guerrouj in ...	Sports	sporting awards	0.072
Plans for new Beagle trip to Mars. The team behind Beagle 2, the failed mission to land on Mars and search for life, have unveiled plans for a successor. Professor Colin Pillinger, lead...	Science	space exploration	0.183

Table 10: Examples of explanations in terms of key phrases with minimum value of f_{ij}^{PPL} for the AGNews dataset.

	SST-2	AGNews
CARP (Few-shot + kNN sampler) (Sun et al., 2023)		
Vanilla	0.940	0.941
CoT	0.955	0.949
CARP	**0.974**	**0.964**
Proposed with GPT-Neo-2.7B		
SVM (both PPL & LL features)	0.890	0.860
LR (both PPL & LL features)	0.890	0.860
Proposed with GPT2-XL		
SVM (both PPL & LL features)	0.920	0.805
LR (both PPL & LL features)	0.920	0.850
Proposed with Falcon-7B-Instruct		
SVM (both PPL & LL features)	0.900	0.860
LR (both PPL & LL features)	0.900	0.830

Table 11: Comparing performance of our approaches using moderate-sized LMs namely GPT-Neo-2.7B, GPT2-XL, and Falcon-7B models against the best approaches from (Sun et al., 2023) which uses the GPT-3

were used to compute perplexity and log-likelihood values, respectively. For the baselines ZS-KP, ZS-KP-CoT, FS-ICL based on these models, we used text-generation pipeline with the temperature parameter as 0.1. The max_tokens parameter was set to 10 for ZS-KP and FS-ICL whereas it was set to 50 for ZS-KP-CoT.

CHT baseline: We used AutoModelForSequenceClassification[6] which adds a classifier head on top of an LM. During training, we tuned only this classifier head (and no other LM parameters) using labelled training examples. The hyperparameters used were: batch_size = 16, #epochs = 30, AdamW optimizer, learning rate=3e-4. For the CHT-BERT baseline based on bert-large-uncased model, we used the following hyperparameters: batch_size = 16, #epochs = 50, AdamW optimizer, learning rate=2e-5. For both CHT and CHT-BERT, the best performing model as per validation accuracy across the epochs was saved and used for evaluation on test set.

SVM: We used the implementation of SVC classifier[7] from the scikit-learn python package, with *linear kernel* and default values for other hyperparameters.

LR: We used the implementation of Logistic Regression classifier[8] from the scikit-learn python package with balanced class weights, maximum number of iterations as 10000, and default values for other hyperparameters.

Multi-label classification: For multi-label datasets - Ethos and Audit reports, we employed One-vs-All strategy where multiple binary classifiers are trained for each label Y to discriminate between Y (positive label) and not-Y (negative label). During inference, more than one label may be predicted for an instance, if more than one binary classifiers predict a positive label (Y). Also, for some instances, no label would be predicted if all of the binary classifiers predict a negative label. For evaluation, we used micro-averaged F1-score computed over all

[6] https://huggingface.co/transformers/v3.0.2/model_doc/auto.html#automodelforsequenceclassification

[7] https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

[8] https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

the labels.

Computing Infrastructure: For running inference with GPT-Neo-2.7B and GPT2-XL (for PPL/LL features computation), we used NVidia A100 GPU with 20GB RAM. For CHT baseline, the same GPU was used. For all experiments related to learning and inference with SVM and LR classifiers, we used a standard laptop with 8GB RAM and Intel i5 processor.

A.5 Ablation Analysis

Table 12 shows the ablation analysis results for the GPT-Neo-2.7B model. Similar to GPT2-XL model (Table 6), the aspects of horizontal scaling, LL features, class-level features and keyphrase-level features, are found to be contributing to achieve the best accuracy. However, the classification results are actually improving in the absence of PPL features for 4 out of 5 datasets. This indicates that using only LL features would be more beneficial in case of GPT-Neo-2.7B model in supervised setting. Though, in zero-shot setting, ZS-PPL performs better than ZS-LL for 3 out of 5 datasets (Table 4).

B Analysis of Audit Reports

In Table 14, we list the classes with a brief description and an example sentence from an audit report for each class.

Description of the ChatGPT Prompt: We use ChatGPT's user interface to perform the classification of the sentences in the test set by prompting it with suitable prompts. The prompt consists of a main instruction, descriptions of the 15 complex classes and finally a set of sentences to classify. The prompt template is shown in Table 13, where text in round brackets is for explanation only. As can be seen, that this is a zero-shot setting of classifying using an LLM. A few shot setting, as part of in-context learning, can also be tried where examples of sentences and their gold class can be provided. However, selection of the classes to give as examples and maintaining the instruction's context are some important challenges, exploration of which we keep as future work.

	SST-2	TREC	AGNews	DBPedia	Ethos
SVM default setting: With all features and horizontal scaling	**0.893**	0.804	0.860	0.912	0.671
SVM default setting without Horizontal scaling	0.882	0.784	0.781	0.915	0.645
SVM default setting without LL features	**0.893**	0.690	0.838	0.877	0.651
SVM default setting without PPL features	0.892	**0.834**	**0.861**	**0.923**	**0.693**
SVM default setting without class-level features	0.892	0.796	0.854	0.918	0.670
SVM default setting without keyphrase-level features	0.842	0.568	0.776	0.902	0.658
LR default setting: With all features and horizontal scaling	**0.893**	0.798	0.858	0.926	0.673
LR default setting without Horizontal scaling	0.890	0.796	0.799	0.917	0.681
LR default setting without LL features	0.885	0.724	0.842	0.893	0.640
LR default setting without PPL features	0.891	**0.812**	**0.861**	**0.932**	**0.686**
LR default setting without class-level features	**0.893**	0.800	0.857	0.924	0.671
LR default setting without keyphrase-level features	0.837	0.558	0.782	0.903	0.660

Table 12: Ablation analysis with the GPT-Neo-2.7B model

(——*Main Instruction*—-)
The task is to classify sentences in a financial audit report into one or more of the following classes. Each line below mentions a class name followed by its description.

(——*Class Descriptions*—-)
1. cost records: About maintenance of cost records.
2. fixed assets: About fixed assets such as equipment, land, building, plant, machinery and their physical verification.
3. human resources and payroll processing: About human resources and payroll processing such as employee wages, leaves, bonus, pension, full and final settlement, policies for leave, gratuity or pension.
4. internal control system: About internal control procedures.
. . .
14. statutory dues: About depositing statutory dues like provident fund, ESI, income tax, sales tax, VAT, service tax, GST, duty of customs, duty of excise.
15. working capital: About working capital, cash credit and bank balance.

(——*Input Sentences for Classification*—-)
What are the applicable classes for the following sentences? Simply print the output as Sentence ID: Class name.
Sentence 1: We have audited the accompanying financial statements of ...
Sentence 2: Management is responsible for the preparation of these financial statements that give a true
. . .
Sentence 10: We conducted our audit in accordance with the Standards on Auditing issued ...

Table 13: ChatGPT Prompt Template

Class	Description	Example Sentence
cost records	A remark about maintenance of cost records.	However, we have not made a detailed examination of the cost records with a view to determine whether they are accurate or complete.
fixed assets	Remarks on purchase of fixed assets, holding of benami property, physical verification of property, plant and equipment by the management at reasonable intervals.	The company has maintained proper records showing full particulars, including quantitative details and situation of fixed assets.
human resources, payroll processing	Remarks on employee wages, leaves, bonus, pension, full and final settlement and mentions of policies for leave, gratuity and pension.	Also Defined benefits obligations in nature of Gratuity and Leave encashment are to be accounted on accrual basis.
internal control system	Remarks on evaluation of internal control procedures with respect to the size and the nature of the company.	During the course of our audit, no major weakness has been noticed in the internal control system in respect of these areas.
inventory	Remarks on possession and purchase of inventory, its physical verification at timely intervals and record keeping	On the basis of the records of inventory, we are of the opinion that the Company is maintaining proper records of inventory and no material discrepancies were noticed on physical verification.
investments	Remarks on investments by the company and compliance to respective Acts	The company has a strategic long term investments in Equity Shares of certain companies, the cost of acquisition of those investments is Rs. 722.50 lacs.
litigations	Remarks about ongoing litigations on the company	Contempt Petition filed against Excise Department at Allahabad High Court against our refund of Rs. 17,25,392/ - against the order of Supreme Court in our favor.
material uncertainty	Remarks on material uncertainties for the company such as net worth, accumulated losses and going concern	The Company ’s accumulated losses at the end of the financial year are less than fifty per cent of its net worth.
operational and administrative expenses	Remarks on company’s operational expenses	The Company has Capitalized expenses to the tune of Rs. 25.40 Crores in Pulp Mill Unit till the date of last balance sheet...
payables	Remarks on details of amount/money to be paid by the company such as repayment of loans	The repayment of loan is on demand, there is no overdue amount remain outstanding.
purchase and procurement	Remarks on purchases and procurement of any kind	The activities of the Company do not involve purchase of inventory and the sale of goods.
receivables	Remarks on details of amount/money to be received by the company such as loans given	The net amount recoverable of Rs. 23640.05 million is subject to reconciliation and confirmation.
sales, services and revenue	Remarks on sales, services and revenue	The Company is a service company, primarily rendering software services.
statutory dues	Remarks on payment of statutory dues and related disputes	The Company is regular in depositing with appropriate authorities undisputed statutory dues including provident fund, employees ’ state insurance ...
working capital	Remarks on working capital and cash/bank balance	No long terms funds have been used to finance short - term except permanent working capital.

Table 14: List of classes in the annotated audit reports with their description and examples

Label	Key phrases
cost records	`cost records`
internal control system	`internal control procedures`
inventory	`inventory`, `physical verification of inventories`
investments	`investments in shares`, `investments in securities`
fixed assets	`fixed assets`, `land or building`, `equipment or machinery`, `physical verification of assets`
human resources and payroll processing	`human resources`, `payroll processing`, `employee wages`, `leave encashment`, `pension or gratuity`
litigations	`litigation`, `court cases`, `appeals at a court or tribunal`
material uncertainty	`erosion of net worth`, `accumulated losses`
operational and administrative expenses	`operational expenses`, `administrative expenses`
purchase and procurement	`purchase of raw materials`, `procurement of raw materials`
payables	`loans taken by the company`, `interest to be paid`, `accepted deposits`, `guarantees given on loans by others`, `repayment of loans`
receivables	`money to be received`, `loans given by the company`
sales, services and revenue	`sale of goods`, `sale of services`, `revenue of the company`
statutory dues	`statutory dues`, `statutory liabilities`
working capital	`working capital`, `cash credit`

Table 15: Key phrases used for the Audit Reports dataset