DA-Pred: Performance Prediction for Text Summarization under Domain-Shift and Instruct-Tuning

Anum Afzal
Technical University of Munich
anum.afzal@tum.de

Florian Matthes
Technical University of Munich
matthes@tum.de

Alexander R. Fabbri
Salesforce AI Research
afabbri@salesforce.com

Abstract

Large Language Models (LLMs) often don't perform as expected under Domain Shift or after Instruct-tuning. A reliable indicator of LLM performance in these settings could assist in decision-making. We present a method that uses the known performance in high-resource domains and fine-tuning settings to predict performance in low-resource domains or base models, respectively. In our paper, we formulate the task of performance prediction, construct a dataset for it, and train regression models to predict the said change in performance. Our proposed methodology is lightweight and, in practice, can help researchers & practitioners decide if resources should be allocated for data labeling and LLM Instruct-tuning.

1 Introduction

Domain adaptation in Large Language Models refers to the ability of such models to understand and generate output for a new domain that is not part of their original training. Typically, LLMs are trained on high-resource domains, and by proxy, LLM performance on those domains is outstanding. Theoretically, LLMs should generalize well to domains not part of their training data. However, the reality is far from this (Basmov et al., 2024). Benchmarks show that when evaluated on a new domain, LLMs, and especially smaller ones, do not perform so well (Afzal et al., 2024).

In practice, LLMs are often fine-tuned before they can be used reliably on a new domain. Such techniques require hardware and domain-specific labeled data. Acquiring these resources can be both expensive and challenging, especially when the new domain in question is also a low-resource one. Furthermore, reliable indicators of how an LLM would generalize across a new domain or under Instruct-tuning approaches are currently missing for Text Summarization. Current performance metrics within common benchmarks for Text Summarization include ROUGE and BERTScore, which require reference summaries. While these metrics can provide insight into LLM performance, they pose limitations against the evaluation of LLMs for low-resource domains, such as the need for significant amounts of annotated data to reveal a domain shift (Van Asch and Daelemans, 2010a). We address this issue by presenting an approach that can predict performance under domain shift for low-resource domains without any labeled data. We first define the task of performance prediction by leveraging the performance indicators of a high-resource domain and the similarity between high-resource and low-resource domains as the basis for prediction.

Over the years, there have been attempts to predict the performance in Classification (Xia et al., 2020; Elsahar and Gallé, 2019; Pogrebnyakov and Shaghaghian, 2021) and Multilinguality (Srinivasan et al., 2021; Patankar et al., 2022). The efforts for performance prediction on text summarization have been rather limited, albeit not completely missing. Louis and Nenkova (2009) formulate it as a classification problem by classifying a document into one of the two classes "good performance" and "poor performance". Similarly, Li et al. (2024) presents an analysis as to which characteristics of the corpus play a role in Text Summarization performance. All previous works have focused on limited domains and used n-gram-based features for analysis and prediction. We build on these previous works by formulating performance prediction as a regression problem, which, to the best of our knowledge, hasn't been done before. We predict the change in performance over a wide range of domains by utilizing a combination of n-gram and contextual metrics.

To summarize, we address the challenge of predicting the performance change of LLMs under domain shift in a setting where no labeled data is available or training resources might be scarce. We

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 7621–7632
November 4-9, 2025 ©2025 Association for Computational Linguistics

construct and release our DA-Pred Dataset[1]. Although already quite diverse, our domain catalog can be easily expanded to include more domains. Our contributions are as follows:

- We introduce a method that leverages known performance on similar high-resource domains to predict performance change on a new low-resource domain for Text Summarization[2]

- We create a dataset for this task using 14 datasets with a wide range of metrics characterizing the datasets.

- We train four Regression models for performance predictions on low-resource domains and present our findings about the importance of including features beyond n-gram overlap.

2 Related Work

Attempting to predict model performance under domain shift has been an ongoing effort. Prior works that attempted to predict model performance include Louis and Nenkova (2009), which posed performance prediction in Automatic Summarization as a classification problem with two prediction classes: "poor performance" and "good performance". Although quite helpful, their work expresses the problem of performance prediction as a binary one and is not applicable to cross-domain prediction. We built on this work by expressing the change in performance as a weighted average for several text summarization evaluation metrics and making it generalizable over domains. Furthermore, Li et al. (2024) investigates the factors that play a role in domain adaptation for text summarization. They characterize aspects like learning difficulty, cross-domain overlap, and word count while using compression ratio and abstraction level to predict the model performance under domain shift. Our work differs in our focus on expanding features and developing models for performance prediction rather than analysis.

The field of performance prediction under domain shift is highly applicable to other tasks as well. Xia et al. (2020) attempts to predict the performance of a language model under domain shift for 9 tasks by training a regression model for each task. However, they leave out complex tasks such as text summarization and question answering. Furthermore, Elsahar and Gallé (2019) proposes methods that use H-divergence, reverse classification accuracy, and confidence measures to predict the performance drop under domain shift for sentiment classification and a sequence labeling task. Alternatively, Pogrebnyakov and Shaghaghian (2021) uses metrics like KL divergence, dataset size, and distribution similarity to predict the success of Domain Adaptation. They leverage the most common source domains in a transfer learning setup for a text similarity task. In another attempt, researchers found a linear relationship between domain similarity and model performance for POS Tagging (Van Asch and Daelemans, 2010b).

Performance prediction has also been explored for Multilinguality in (L)LMs. Srinivasan et al. (2021) provides a holistic overview of performance prediction in a multilingual setting. They employ a regression model that leverages various syntactical characteristics and performance scores of a similar language to predict performance on an unseen language. A similar effort is made by Patankar et al. (2022), who use several regression models to predict the performance of Multilingual LLMs for the SumEval 2022 shared task. On the other hand, Khiu et al. (2024) used domain similarity to predict performance on low-resource languages for machine translation. They use features such as the size of the Instruct-tuning corpus, the domain similarity between Instruct-tuning and testing corpora, and the language similarity between source and target languages as the basics for prediction. They report domain similarity to have a big impact on performance.

3 Domains

To make our methodology generalizable, we utilize 14 text summarization datasets spanning six domains: Medical, Science, Law, Government, News, and Conversation. Table 1 provides a summary of the average article and summary lengths for each dataset by domain. Although the number of datasets per domain varies, the datasets within each domain are diverse in origin, which helps address potential concerns about class imbalance in our experiments. The datasets were chosen based on their availability and strong representation in previous work on domain adaptation for text summarization.

[1] Code and Datasets for DA-Pred can be found at github.com/anum94/DAPred

[2] DA-Pred could be easily adapted for other tasks by replacing the text summarization evaluation metrics with the ones suitable for the other task.

3.1 Medical

We used **PubMed** (Cohan et al., 2018), which contains articles from the medical domain along with their summaries, and **Lay Summarization** by Goldsack et al. (2022), which contains Biomedical articles and their expert-written lay summaries. Contrary to PubMed, the summaries in this dataset are layman summaries of the rather technical articles.

3.2 Science

We include **ArXiv** (Bhattacharya and Getoor, 2007), derived from the ArXiv OpenAccess repository, and **ACLSum** (Takeshita et al., 2024) consists of Natural Language Processing research papers and summaries of these articles manually written by domain experts.

3.3 Law

Our suite includes **Big Patent**[3] (Sharma et al., 2019) contains U.S. patent documents along with human written summaries, and **Multi-LexSum** dataset (Shen et al., 2022) consists of legal case summaries as articles followed by expert-authored summaries.

3.4 Government

We include the **GovReport** dataset by Huang et al. (2021), which is a collection of long reports published by the U.S. Government Accountability Office, and **BillSum** dataset (Kornilova and Eidelman, 2019), which uses a collection of U.S. Congress and California state bills as an article along with a human-written summary from the Congressional Research Service.

3.5 News

Our news domains consists of four datasets including **Newsroom** by Grusky et al. (2018) contains the news snippet as the article and reference summaries written by domain experts, **CNN / DailyMail** dataset (Nallapati et al., 2016) consisting of news article and bullet point highlights as summaries, **Gigaword** (Graff et al., 2003) which follows the same structure as CNN / DailMail, and lastly **XL-Sum** dataset (Hasan et al., 2021) contains news article and their respective summaries from BBC.

3.6 Conversation

The Conversation domain consists of two datasets, including **SamSum** dataset (Gliwa et al., 2019), which is manually generated by linguists, and each messenger-like conversation contains a summary of the topic discussed, and **DialogSum** dataset (Chen et al., 2021), which is similar in nature to SamSum

Domain	Dataset	#W	#W Sum
Medical	PubMed	4400	394
	Lay Summarization	26446	969
Science	ACLSum	5779	480
	ArXiv	7414	402
Law	BigPatent	12362	366
	Multi-LexSum	74977	1639
Government	BillSum	5450	318
	GovReport	28285	3221
News	CNN/DM	1108	137
	Newsroom	324	13
	Gigaword	128	31
	XLSum	669	71
Conversation	DialogSum	384	67
	SamSum	338	76

Table 1: Average token count in articles (#W) and Average token count in Summaries (#W Sum) computed on the test split of all datasets computed using `Llama 3.1 Instruct (8b)` tokenizer.

4 DA-Pred Suite

In this section, we explain our performance prediction tasks, dataset construction methodology, and the experimental settings used.

4.1 Performance Prediction Task

Our goal is to predict performance change when a) transitioning from a high-resource domain (source) to a low-resource domain (target), and b) performance gain when transitioning from a base model to a fine-tuned model under the same domain. To evaluate the said change, we study three Domain Adaptation scenarios:

- **No Domain Shift (ND):** There is no domain-shift between source and target.
- **In Domain Shift (ID):** There is a shift in distribution such that the source and target belong to different datasets or different splits of the same dataset.

[3] Although there are 9 subcategories of this dataset, we focus on the category Fixed Constructions.

- **Out of Domain (OOD):** There is an evident domain shift such that the source and target belong to different domains.

We simulate ND, ID, and OOD settings using the datasets explained in section 3. Some examples of dataset sampling under different scenarios are shown in Table 2. For instance, we predict the potential Instruct-tuning performance on the Newsroom dataset based on known Instruct-tuning results from the Gigaword dataset. This models an in-distribution (ID) scenario, where similarities between the datasets and prior performance on Gigaword are used to estimate expected performance changes.

Source		Target		DA
dataset	split	dataset	split	
PubMed (IT)	test	CNN/DM (IT)	test	OOD
ArXiv	train	ArXiv	test	ND
Gigaword (IT)	train	Newsroom (IT)	test	ID
SamSum	test	ArXiv	test	OOD
XLSum (IT)	test	Newsroom (IT)	test	OOD
Multi-LexSum	test	BigPatent	test	ID

Table 2: An illustration of possible samples of the DA-pred dataset where ND, ID, and OOD refer to No Domain Adaptation, In-domain, and Out of Distribution settings. IT refers to inference on the fine-tuned model.

4.2 Dataset Construction

We use the methodology illustrated in Figure 1 to construct our DA-Pred dataset consisting of 392[4] samples. One sample in our dataset represents the performance change for one of the domains under one of the domain adaptation settings. Each sample is constructed using a pair (source domain A, target domain B) that reflects the change in performance when transitioning to target domain B or moving from the base to the fine-tuned model for the same domain. We employ task-specific and task-agnostic metrics as training features and model performance changes as the y_drop to be predicted. The pairs are created by sampling from the 14 datasets shown in Table 2 under 3 domain adaptation settings for both zero-shot and fine-tuning.

As illustrated in Figure 1, we compute domain-similarity metrics between the source and target domains, as well as domain-specific metrics for the target domain. We refer to both of these as task-agnostic metrics[5]. In addition, we compute task-specific metrics for both the source and target domains, but only use the source domain metrics as training features.

There are many available metrics for evaluating text, each with its own strengths and limitations. Since no single metric is reliably sufficient, we follow recent literature (Afzal et al., 2024) on Text Summarization evaluation, which advocates for using multiple metrics to assess performance. To strike a balance between traditional and contemporary approaches, we selected representative metrics from several categories: BERTScore for contextual similarity, ROUGE for n-gram overlap, FActScore for factual consistency, and GPT-4-based evaluations for coherence, relevance, and other subjective aspects, representing the LLM-as-a-judge paradigm. Lastly, task-specific metrics from both the target and the source domains are used to construct the y_drop as follows:

$$y\_drop = y\_weighted_{source} - y\_weighted_{target}$$

$$y\_weighted = \frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i=1}^{n} w_i}$$

$$\begin{aligned} \text{where: } x_i &= \text{value of the } i\text{-th metric} \\ w_i &= \text{weight of the } i\text{-th metric} \\ n &= \text{\# task-specific metrics} \end{aligned}$$

These weighted scores, which aggregate various evaluation metrics, represent model performance on a given domain or post fine-tuning. Accordingly, y_drop ranging from -1 to +1 captures the change in performance, specifically, the drop or gain in quality, when moving from the source to the target domain. In our methodology, the negative and positive values represent the performance drop and gain, respectively. Details on the metrics used in constructing the DA-Pred dataset are provided below. (See Appendix A for details).

Domain Similarity Metrics

- **Vocabulary Overlap:** We compute vocabulary overlap as described by Afzal et al. 2024; Yu et al. 2021 to encapsulate the word overlap between the vocabulary of two given datasets.

- **TF-IDF-weighted Overlap:** This is an adaptation of the Vocabulary Overlap where

[4] When training a regression model, the One in ten rule can be applied. We use only 11 features under feature selection; therefore, our dataset is of sufficient size

[5] We call them task-agnostic because they are computed from the raw corpus and do not require labeled data.

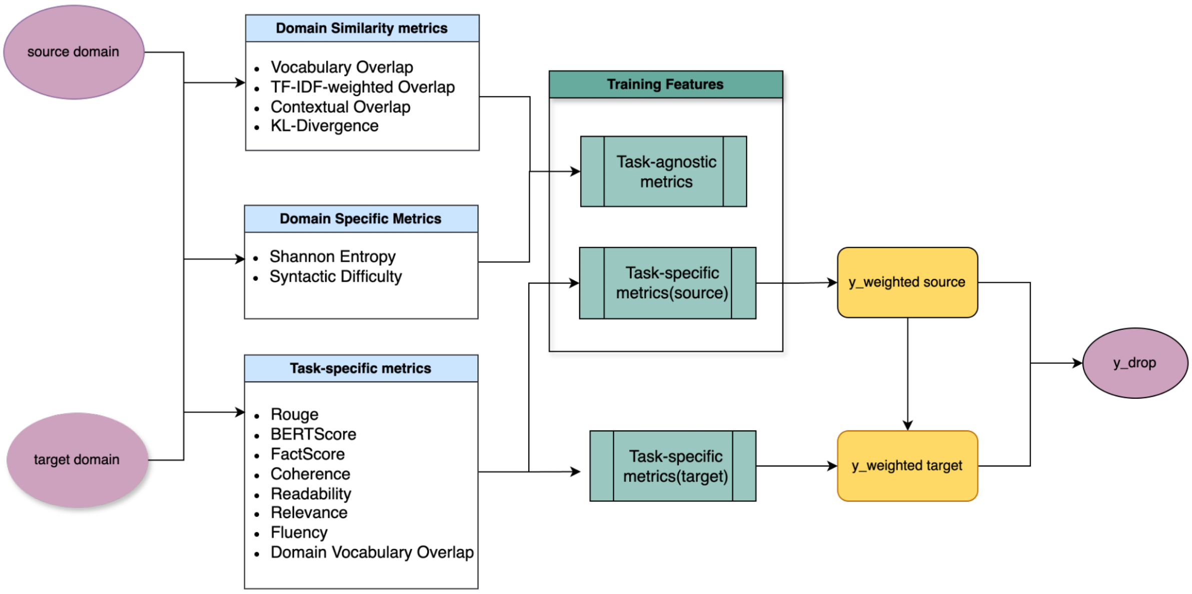

Figure 1: A block diagram depicting the pipeline used to generate the DA-Pred dataset. Each sample is a pair of high-resource (source) and simulated low-resource (target) domains and follows the pipeline shown.

TF_IDF is used to find the top 10k words in both datasets, which is used to compute the overlap. We use `TfidfVectorizer` from `Scikit-learn` (Pedregosa et al., 2012) to assign an importance score to words by providing the list of all articles (documents) without stopwords.

- **Contextual Similarity:** First, we generate embeddings of all articles through a sentence embedding model. Then we compute Cosine Similarity between the embeddings of individual documents and use the average cosine similarity values to describe the contextual similarity between them.
- **Kullback–Leibler (KL) Divergence:** We include KL divergence to capture the similarity or rather dissimilarity between the distribution of two corpora. For each dataset, we devise a probability distribution of the words used in all articles. These distributions are used to compute the KL Divergences between two given datasets.

Domain-Specific Metrics

- **Syntactic Difficulty:** Inspired by the Syntactic Simplicity introduced by Redmiles et al. 2019; Leroy and Endicott 2012, we compute syntactic difficulty by taking a weighted average of Dependency Length and Tree Depth of the corpus. We compute syntactic difficulty by taking a weighted average of the Dependency Length and Tree Depth of the corpus. Dependency length measures the distance between a word and its syntactic head (e.g., the relationship between a verb and its object). Whereas the tree depth is the maximum number of hierarchical levels in a sentence's dependency tree. The syntactic difficult co-efficient σ is calculated as:
$\sigma = \alpha \times avg\_dependency\_length + \beta \times max\_tree\_depth$ where $\alpha = 0.5, \beta = 0.5$
- **Shannon Entropy:** We calculate Shannon Entropy to encapsulate how much new information is present in the dataset. Shannon Entropy is calculated by creating a probability distribution of the dataset by using all words in the articles. We hope that this serves as an indicator of how difficult it might be for LLM to understand this domain.

Task-Specific Metrics

- **ROUGE:** We conduct reference-based evaluation through n-gram by computing the ROUGE score (Lin, 2004). We use ROUGE-1, ROUGE-2, and ROUGE-L in our feature space.
- **BERTScore:** We include the contextual overlap of generated summaries with reference summaries through BERTScore (Zhang et al.,

	All Features				Feature Selection			
	MSE	MAE	$RMSE$	R^2	MSE	MAE	$RMSE$	R^2
ROUGE (baseline)								
Regression	0.05	0.04	0.23	0.41	0.04	0.05	0.23	0.34
Ridge	0.05	0.04	0.23	0.41	0.04	0.04	0.02	0.34
Lasso	0.07	0.06	0.27	0.01	0.06	0.06	0.27	0.011
XGBoost	0.05	0.04	0.22	0.56	0.04	0.05	0.22	0.47
DA-Pred								
Regression	0.03	0.03	0.19	0.92	0.04	0.03	0.21	0.91
Ridge	0.04	0.03	0.19	0.92	0.03	**0.03**	**0.19**	**0.89**
Lasso	0.15	0.11	**0.11**	0.39	0.15	0.11	0.39	0.11
XGBoost	0.03	0.02	0.18	0.94	**0.03**	**0.02**	0.18	0.97

Table 3: The performance of Linear Regression Models through Mean Absolute Error (MAE), Root Mean Squared Error ($RMSE$), and R^2 on the DA-Pred Dataset using K-fold cross validation.

2020). We include BERTScore Precision, BERTScore Recall, and BERTScore F1 score.

- **FActScore:** We also include Factuality as one of the training features and use FActscore (Min et al., 2023) with `GPT-4o-mini` for factuality evaluation.

- **LLM-based Evaluation:** We employ `GPT-4o-mini` as a judge to do a prompt-based evaluation of all generated summaries against Coherence, Readability, Relevance, and Fluency. Our evaluation prompts are inspired by G-Eval (Liu et al., 2023).

- **Domain Vocabulary Overlap:** We construct domain vocabulary by taking the top 10k words of the domain and computing the overlap of the words in the generated summaries with the domain vocabulary.

4.3 Experimental Settings

In this section, we discuss the experimental settings we used to realize the Performance Prediction task using the DA-Pred dataset. For our experiments, we construct our dataset and associated features using `Llama 3.1 Instruct (8b)` (Grattafiori et al., 2024) as the backbone. We run inference for each dataset using `Llama 3.1 Instruct (8b)` and fine-tuned `Llama 3.1 Instruct (8b)` for each dataset. All metrics are calculated using 500 samples from the test set. For contextual similarity, we use OpenAI's *text-embedding-3-small* model. See Appendix A for technical details.

4.4 Prediction Models

Since our training dataset is medium-sized, we employ models that support training on such a dataset through regularization. We include a simple *Linear Regression model* as the baseline and compare it with *Lasso Regression*, *Ridge Regression* and finally an *XGBoost Model* (Chen and Guestrin, 2016). We use Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error ($RMSE$), and Coefficient of Determination (R^2) to evaluate the performance of the prediction models.

4.5 Feature Selection

We use Feature Selection using `scikit-learn`'s `SelectKBest` to select the top-k features based on their variance, such that $k = \sqrt{n}$ where n = the number of training samples. This strategy helps us understand which features are actually important for the prediction of Text Summarization performance and may also help improve model performance.

4.6 ROUGE vs DA-Pred

ROUGE: In previous work (Li et al., 2024), n-gram ROUGE was used as a metric for evaluation of the generated summaries, and hence the performance prediction was the change in ROUGE scores. For this baseline implementation, we keep only the n-gram-based metrics, including Vocabulary Overlap, TF-IDF-weighted Vocabulary Overlap, KL-Divergence, Shannon Entropy, Syntactic Difficulty, and ROUGE as the training features.

DA-Pred: Given the surge of LLM-based evaluation metrics, we included them in our prediction metrics. Contrary to the baseline, which used n-gram as the sole indicator of performance, we use a diverse range of metrics introduced in subsection 4.2 as training features.

5 Results & Discussion

5.1 Prediction Models

As shown in Table 3, we observe that a simple Linear Regression model is not a suitable choice for the performance prediction task, and Lasso fails to learn the characteristics of our dataset. The very high R^2 and very low error scores of XGBoost suggest overfitting. Using a combination of all scores, Ridge Regression seems to be a suitable choice for this task, and we would continue to use it in the rest of our experiments. (See subsection 5.3 for effects of number of datasets)

5.2 ROUGE vs DA-Pred

We use n-gram-based metrics in our baseline method and a combination of n-gram and contextual metrics as training features. Our experiments, as depicted in Table 3, show better test performance with the training features of the DA-pred dataset. The low R^2 scores on the baseline dataset suggest that the n-gram-based features are not enough for the model to comprehend the domain fully. Whereas, for DA-Pred, the low MAE, MSE, $RMSE$ scores, and a high R^2 score complement each other and eliminate the doubts of overfit.

5.3 Effects of Dataset Size

We evaluated the $RMSE$ and R^2 with respect to the number of datasets and show them in Figure 3. Apart from minor fluctuations, it can be seen that the $RMSE$ for all models and both datasets slowly decreases as we increase the number of samples, suggesting better learning. We do not see any improvements in R^2 scores of all models when changing the size of the train set.

5.4 Feature Selection

Baseline		DA-Pred	
Feature	# Selection	Feature	# Selection
vocab_overlap	6	source_coherence	12
source_rouge1	6	source_consistency	12
source_rouge2	6	source_relevance	11
source_rougeL	6	source_factscore	10
target_shannon_entropy	5	source_rouge1	10
tfidf_overlap	4	source_rouge2	8
learning_difficulty	3	source_fluency	6
source_shannon_entropy	1	source_rougeL	5

Table 4: Most important features for the prediction of Text Summarization performance along with their counts.

To better understand which metrics contribute to performance prediction under domain adaptation, we perform feature selection on both the baseline and DA-Pred datasets, as shown in Table 3. Overall, feature selection does not lead to significant changes in model performance, but we still use it as an analytical tool to gain insights into the datasets.

We track the features selected across various values of N, where N is the number of datasets, and count how often each feature is selected. These counts are presented in Table 4. Our analysis reveals that features such as coherence, consistency, relevance, and factual correctness are especially important for performance prediction—yet they are absent in the baseline method.

6 Practical Implications

Our models can be used off-the-shelf to estimate performance on a low-resource dataset by leveraging a related high-resource dataset. As illustrated in Figure 2, performance on similar high-resource datasets within a given domain is assumed to be known in advance. Task-agnostic metrics for the low-resource domain can be computed efficiently, without requiring a GPU. Based on these metrics, our method predicts model performance on the low-resource domain. We created four holdout datasets, each consisting of 1,000 samples, to evaluate our methodology. These include ACLSumm+, which consists of research papers scraped from ACL 2024 proceedings, medical articles scraped from BMJ Medical Research[6], the legal cases from India and UK (Shukla et al., 2022), and lastly, we scraped the articles of current affairs from Q4 2024.

6.1 Human Evaluation

To validate the implications and effectiveness of y_{drop} (ranging from -1 to +1), we compared our model with human evaluators. We generated summaries using zero-shot `LLaMA 3.1 Instruct (8B)` and also the fine-tuned[7] variant of the same model. For each sample, annotators assigned scores in two evaluation settings when transitioning from 1) a high-resource domain summary to a similar low-resource domain summary, and 2) a zero-shot summary to an Instruct-tuned summary, within the same domain.

We recruited eight annotators from a local university and compensated them at 16 euros per hour. All annotators held graduate degrees in the domain

[6] `https://www.bmj.com/research/research`

[7] We used `GPT-4.1` to generate summaries for Holdout datasets for Instruct-tuning

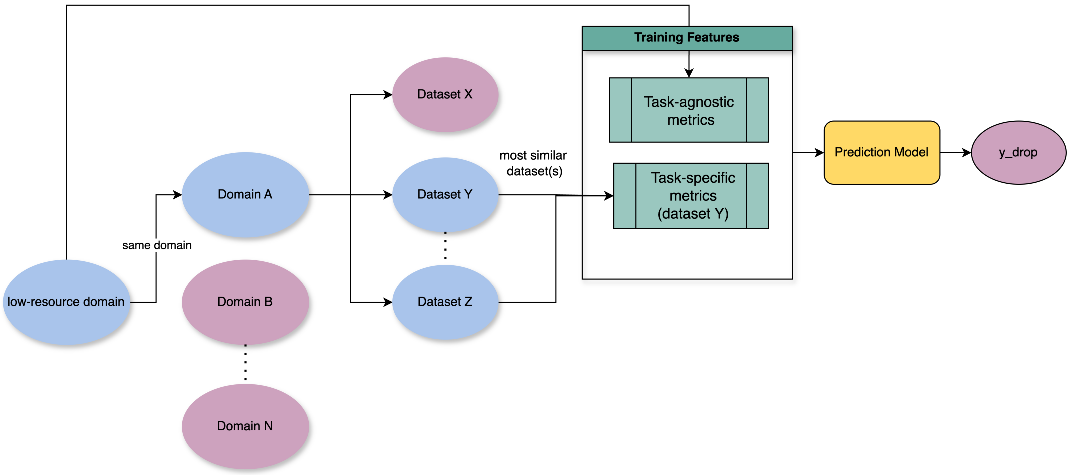

Figure 2: The lightweight methodology used to obtain predictions on a new domain. The illustration uses 2 domains, but the number of domains can be configured.

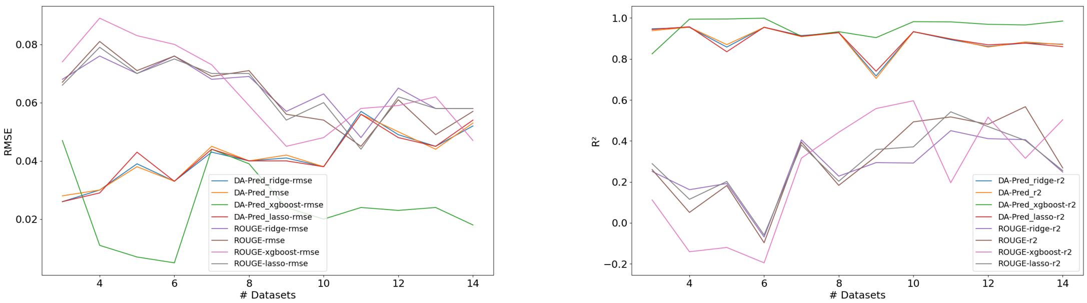

Figure 3: $RMSE$ and R^2 scores of models recorded when trained over N samples where N = number of datasets

they were evaluating. Each sample was annotated by two annotators. Each annotator was asked to assign a score between -1 and +1, reflecting the quality of a summary relative to a provided reference. The inter-annotator agreement, measured using Pearson correlation, was 0.4. We report the average y_drop assigned by human evaluators and predicted by our method on 25 samples in Table 5.

6.2 Mapping y_drop to Recommended Actions

To further support practitioners, we used annotator feedback to map y_drop values to practical recommendations. For each sample, three summaries were generated—one from each of the zero-shot, in-context learning (ICL), and Instruct-tuned settings. Annotators were shown a reference summary from a similar high-resource domain, along with all three generated summaries for the corresponding sample in the holdout dataset. They first assigned a score between -1 and +1 to the zero-shot summary. Then, after reviewing the ICL and Instruct-tuned summaries, they were asked to assign a recommendation label based on their comparative assessment. We present the mapping between y_ drop and these recommendation labels in Figure 4. The recommendation labels were:

Fine-tuned Summary preferred: if the annotator preferred the Instruct-tuned summary,

ICL Summary preferred: if they preferred the ICL summary,

zero-shot summary preferred: if neither the ICL nor the instruct-tuned summaries were judged better.

7 Conclusion and Future Work

We introduced a Performance Prediction task for Text Summarization under Domain Shift and Instruct-Tuning. We created a dataset for it covering 6 domains and 14 datasets under various

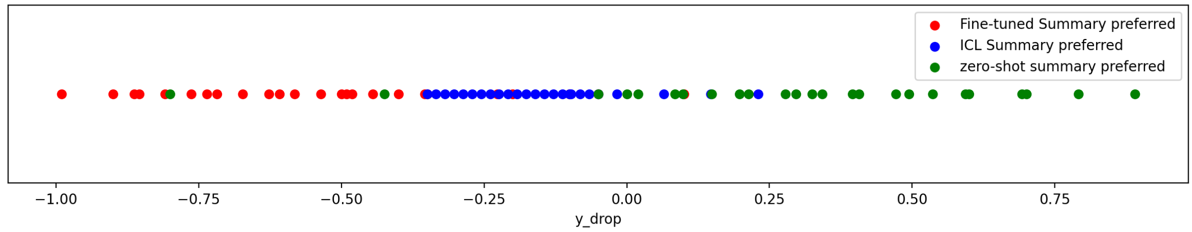

Figure 4: We show the y_drop scores assigned by human Annotator along with their recommended actions after seeing the In-context Learning summary and Instruct-tuned summary of our 4 Holdout Validation datasets. Major Improvements, Minor Improvements, and no Improvements needed imply *Instruct-tuning*, *ICL*, and *use as is*, respectively.

Holdout Dataset	Similar Datasets	DA∇	H∇
ACLSum	ArXiv	0.02	0.1
Scrapped News	Newsroom	0.07	0
Scrapped Legal	BigPatent	-0.23	-0.3
BMJ Medical Research	PubMed	-0.86	-0.9

Table 5: DA∇ and $H\nabla$ are the change in performance depicted by our method and human annotator, respectively.

domain adaptation settings for zero-shot and fine-tuned models. We train four prediction models for this task using the DA-Pred dataset. Our experiments show a Ridge Regression model with DA-pred features to be most suitable, whereas Feature Selection shows only minor improvement. Through human evaluation, we show our method to be reliable and also provide a mapping of y_drop to recommended actions for practitioners. Text Summarization is a complex task that evaluates an LLM's Text Modeling ability on a wide spectrum. Based on this motivation and the existing research gap in Text Summarization performance prediction, we selected it for evaluating our methodology. However, our approach can be adapted to predict performance for all tasks under domain shift by simply replacing the task-specific metrics. We introduce task-agnostic metrics briefly in the paper, but the broader implication is to include metrics like Perplexity that capture LLM competence on a domain without relying on task-specific metrics and potentially evolving towards task-agnostic performance prediction for LLM under domain shift.

Limitations

Due to the high computational effort needed for the construction of each sample in the dataset, we limited the metrics computation to 500 samples per dataset. This sample size, however, is still large enough to be statistically significant.

The task introduced in our paper focuses on the change in performance when switching from one domain to the other, and not so much on the performance change when switching models. We use only one model, such as `Llama 3.1 Instruct (8b)` as an experimental choice for computing task-specific scores needed to validate the methodology. Switching to Mistral, for example, should not impact how the regression models learn to comprehend the domains. The dataset introduced in this paper is small in comparison to datasets that are nowadays used to train deep learning models, although still sufficient for training a light-weight Regression Model. Including more models in the DA-pred suite could help increase the size of the training data.

Lastly, our experiments are designed to simulate low-resource domains in which we do have Instruct-tuned models / labeled data available. We use existing High-Resource datasets to simulate a real-world low-resource setting.

Ethical Statement

We present a method for the performance prediction of Text Summarization under domain shift to foster research in this area. In our dataset construction, we used open-source Text Summarization datasets. Our human annotation guidelines were carefully crafted, and annotators were compensated. We performed Instruct-tuning also using the open-source datasets and thus did not include any bias in the model other than what might already be part of the model/datasets.

References

Anum Afzal, Ribin Chalumattu, Florian Matthes, and Laura Mascarell. 2024. AdaptEval: Evaluating large language models on domain adaptation for text summarization. In *Proceedings of the 1st Workshop on Customizable NLP: Progress and Challenges in Customizing NLP for a Domain, Application, Group, or Individual (CustomNLP4U)*, pages 76–85, Miami, Florida, USA. Association for Computational Linguistics.

Victoria Basmov, Yoav Goldberg, and Reut Tsarfaty. 2024. Llms' reading comprehension is affected by parametric knowledge and struggles with hypothetical statements. *Preprint*, arXiv:2404.06283.

Indrajit Bhattacharya and Lise Getoor. 2007. Collective entity resolution in relational data. *ACM Transactions on Knowledge Discovery from Data*, 1(1):1–36.

Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 785–794. ACM.

Yulong Chen, Yang Liu, Liang Chen, and Yue Zhang. 2021. DialogSum: A real-life scenario dialogue summarization dataset. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 5062–5074, Online. Association for Computational Linguistics.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A discourse-aware attention model for abstractive summarization of long documents. *Preprint*, arXiv:1804.05685.

Hady Elsahar and Matthias Gallé. 2019. To annotate or not? predicting performance drop under domain shift. 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)*, pages 2163–2173, Hong Kong, China. Association for Computational Linguistics.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 70–79, Hong Kong, China. Association for Computational Linguistics.

Tomas Goldsack, Zhihao Zhang, Chenghua Lin, and Carolina Scarton. 2022. Making science simple: Corpora for the lay summarisation of scientific literature. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10589–10604, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2003. English gigaword. *Linguistic Data Consortium, Philadelphia*, 4(1):34.

A. Grattafiori, A. Dubey, A. Jauhri, et al. 2024. The llama 3 herd of models. *arXiv*, abs/2407.21783.

Max Grusky, Mor Naaman, and Yoav Artzi. 2018. Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 708–719, New Orleans, Louisiana. Association for Computational Linguistics.

Tahmid Hasan, Abhik Bhattacharjee, Md Saiful Islam, Kazi Samin, Yuan-Fang Li, Yong-Bin Kang, M. Sohel Rahman, and Rifat Shahriyar. 2021. Xl-sum: Large-scale multilingual abstractive summarization for 44 languages. *Preprint*, arXiv:2106.13822.

Luyang Huang, Shuyang Cao, Nikolaus Parulian, Heng Ji, and Lu Wang. 2021. Efficient attentions for long document summarization. *Preprint*, arXiv:2104.02112.

Eric Khiu, Hasti Toossi, David Anugraha, Jinyu Liu, Jiaxu Li, Juan Flores, Leandro Roman, A. Seza Doğruöz, and En-Shiun Lee. 2024. Predicting machine translation performance on low-resource languages: The role of domain similarity. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 1474–1486, St. Julian's, Malta. Association for Computational Linguistics.

Anastassia Kornilova and Vladimir Eidelman. 2019. BillSum: A corpus for automatic summarization of US legislation. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 48–56, Hong Kong, China. Association for Computational Linguistics.

Gondy Leroy and James E. Endicott. 2012. Combining nlp with evidence-based methods to find text metrics related to perceived and actual text difficulty. In *Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium*, IHI '12, page 749–754, New York, NY, USA. Association for Computing Machinery.

Yinghao Li, Siyu Miao, Heyan Huang, and Yang Gao. 2024. Word matters: What influences domain adaptation in summarization? In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13236–13249, Bangkok, Thailand. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval:

NLG evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522, Singapore. Association for Computational Linguistics.

Annie Louis and Ani Nenkova. 2009. Performance confidence estimation for automatic summarization. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 541–548, Athens, Greece. Association for Computational Linguistics.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100, Singapore. Association for Computational Linguistics.

Ramesh Nallapati, Bowen Zhou, Cicero Nogueira dos santos, Caglar Gulcehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. *Preprint*, arXiv:1602.06023.

Shantanu Patankar, Omkar Gokhale, Onkar Litake, Aditya Mandke, and Dipali Kadam. 2022. To train or not to train: Predicting the performance of massively multilingual models. In *Proceedings of the First Workshop on Scaling Up Multilingual Evaluation*, pages 8–12, Online. Association for Computational Linguistics.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake VanderPlas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Edouard Duchesnay. 2012. Scikit-learn: Machine learning in python. *CoRR*, abs/1201.0490.

Nick Pogrebnyakov and Shohreh Shaghaghian. 2021. Predicting the success of domain adaptation in text similarity. In *Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP-2021)*, pages 206–212, Online. Association for Computational Linguistics.

Elissa Redmiles, Lisa Maszkiewicz, Emily Hwang, Dhruv Kuchhal, Everest Liu, Miraida Morales, Denis Peskov, Sudha Rao, Rock Stevens, Kristina Gligorić, Sean Kross, Michelle Mazurek, and Hal Daumé III. 2019. Comparing and developing tools to measure the readability of domain-specific texts. 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)*, pages 4831–4842, Hong Kong, China. Association for Computational Linguistics.

Eva Sharma, Chen Li, and Lu Wang. 2019. BIGPATENT: A large-scale dataset for abstractive and coherent summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2204–2213, Florence, Italy. Association for Computational Linguistics.

Zejiang Shen, Kyle Lo, Lauren Yu, Nathan Dahlberg, Margo Schlanger, and Doug Downey. 2022. Multi-lexsum: Real-world summaries of civil rights lawsuits at multiple granularities. *CoRR*, abs/2206.10883.

Abhay Shukla, Paheli Bhattacharya, Soham Poddar, Rajdeep Mukherjee, Kripabandhu Ghosh, Pawan Goyal, and Saptarshi Ghosh. 2022. Legal case document summarization: Extractive and abstractive methods and their evaluation. *Preprint*, arXiv:2210.07544.

Anirudh Srinivasan, Sunayana Sitaram, Tanuja Ganu, Sandipan Dandapat, Kalika Bali, and Monojit Choudhury. 2021. Predicting the performance of multilingual nlp models. *Preprint*, arXiv:2110.08875.

Sotaro Takeshita, Tommaso Green, Ines Reinig, Kai Eckert, and Simone Ponzetto. 2024. ACLSum: A new dataset for aspect-based summarization of scientific publications. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6660–6675, Mexico City, Mexico. Association for Computational Linguistics.

Vincent Van Asch and Walter Daelemans. 2010a. Using domain similarity for performance estimation. In *Proceedings of the 2010 Workshop on Domain Adaptation for Natural Language Processing*, pages 31–36, Uppsala, Sweden. Association for Computational Linguistics.

Vincent Van Asch and Walter Daelemans. 2010b. Using domain similarity for performance estimation. In *Proceedings of the 2010 Workshop on Domain Adaptation for Natural Language Processing*, pages 31–36, Uppsala, Sweden. Association for Computational Linguistics.

Mengzhou Xia, Antonios Anastasopoulos, Ruochen Xu, Yiming Yang, and Graham Neubig. 2020. Predicting performance for natural language processing tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8625–8646, Online. Association for Computational Linguistics.

Tiezheng Yu, Zihan Liu, and Pascale Fung. 2021. AdaptSum: Towards low-resource domain adaptation for abstractive summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5892–5904, Online. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. *Preprint*, arXiv:1904.09675.

A Experimental Settings

A.1 DA-Pred Dataset

For Inference on Instruct-tuned models, we use an A100 80GB, whereas for zero-shot inference, we use API endpoints provided by Together AI[8]. We Instruct-tuned `Llama 3.1 Instruct 8B` on 10k samples from the train split with a context window of 8192 tokens. We trained for 2 epochs with a batch size of 64 using the Cerebras Framework[9]. Unless specified, we use the default values for all models and methods.

A.2 Prediction Models

We train all Regression models with default settings and employ RidgeCV and LassoCV provided by `Scikit-learn` to automatically select the best values of α. For XGBoost, we train it for 20 epochs and pick the best model based on the validation loss. For better training, we normalize all our features to be on a scale of 0 - 1 and use K-fold cross-validation with K=5.

[8] https://api.together.ai/

[9] https://docs.cerebras.ai/